Ultra Packet Core CUPS User Plane Administration Guide, Release 21.26

First Published: 2021-12-24

Last Modified: 2023-02-03

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2021-2023 Cisco Systems, Inc. All rights reserved.

CONTENTS

PREFACE

About this Guide xliii

Conventions Used xliii

CHAPTER 1

Overview 1

Product Description 1

Supported Features and Functionality 3

3GPP ULI Enhanced Reporting Support 3

AAA Server Group 3

APN Configuration Support 3

Asynchronous Core Transfer Support for egtpinmgr 4

Charging Data Records to HDD 5

GTP-C Path Failure Enhancements and Improved Debugging Tools 5

GTPP Suppress-CDR No Zero Volume 6

Location Based DNS and PCSCF IP Address Selection 6

MPRA Support 7

No udp-checksum Support 7

QUIC IETF Implementation 8

Configuring QUIC IETF 8

Optimization for egtpinmgr Recovery 8

Quota Hold Time Support 8

S-GW Paging Enhancement 9

Session Recovery in User Plane 10

SRVCC PS to CS Handover Indication and the QoS Class Index IMS Media Configuration Support 10

Support for ip hide-service-address CLI Command 12

Support for regardless-of-other-triggers CLI Command 12

TFT Suppression for Default Bearer 13

Feature Description 13
Configuring TFT Suppression 13
Zero-byte EDR Suppression 14
How It Works 15
Call Flows 15
P-GW Data Session 15
S-GW Data Session 16
Support for Addition, Deletion and Updation of Dedicated Bearers for S-GW 19
Support for Collapse Call 20
P-GW Session Reporting with Gy Interface 24
P-GW Session Reporting with Gz Interface 33
Bit Rate Mapping Support 35
Standards Compliance 36

CHAPTER 2 **Configuring User Plane in CUPS 37**
Configuring User Plane Service 37
Associating GTP-U Service with User Plane Service 38
Associating Sx Service to User Plane Service 39
Recommended Timers 39
Recommended Configurations 40
Example Configurations in CP 40
Example Router Configurations 43
Example Configurations in UP 44
Example SRP Configurations 45

CHAPTER 3 **Monitoring and Troubleshooting User Plane in CUPS 47**
Monitoring and Troubleshooting User Plane in CUPS 47
SNMP Traps 47
Show Commands 48
show configuration 48
show-gtpu-statistics 48
show module p2p user-plane-ipv6-addr 51
show saegw-service all 51
show saegw-service name 51

show service all 51
show subscriber all 51
show subscribers user-plane-only all 52
show subscribers user-plane-only called/seid called/seid flow flow-id flow-id 52
show subscribers user-plane-only called/seid called/seid flows full 53
show subscribers user-plane-only called/seid called/seid flows 54
show subscribers user-plane-only callid call_id pdr all 54
show subscribers user-plane-only callid/seid callid/seid pdr full all 54
show subscribers user-plane-only callid/seid callid/seid pdr id pdr-id 56
show subscribers user-plane-only flows 57
show subscribers user-plane-only full all 58
show subscribers user-plane-only seid seid pdr all 60
show user-plane-service [all | name name] 60
show user-plane-service statistics all 61
show user-plane-service statistics charging action 65
show user-plane-service statistics group-of-ruledefs 67
show user-plane-service statistics ruledef 68

CHAPTER 4 **1:1 User Plane Redundancy for 4G CUPS 69**
Revision History 69
Feature Description 69
How it Works 69
Configuring 1:1 User Plane Redundancy for 4G CUPS 79
Configuring BFD Monitoring Between Active UP and Standby UP 79
Flagging BGP Monitoring Failure 80
Configuring Sx Monitoring on the Active UP and Standby UP 81
Configuring SRP over IPSec on the Active UP and Standby UP 81
Configuring VPP Monitor on Active UP and Standby UP 83
Preventing User Plane Switchback 83
Preventing Dual Active Error Scenarios 84
Resetting Sx Monitor Failure 84
Monitoring and Troubleshooting 84
Show Command(s) and/or Outputs 84
show srp monitor bfd 85

show srp monitor bgp 85
show srp monitor sx 85
show srp monitor vpp 85

CHAPTER 5 5G NSA for SAEGW in CUPS 87
Feature Description 87

CHAPTER 6 Access Control Lists 89
Revision History 89
Feature Description 89
Configuring Access Control Lists 89
Monitoring and Troubleshooting 90
Show Command(s) and/or Outputs 90
show sub user-plane-only full all 91

CHAPTER 7 ADC Over Gx 93
Feature Description 93
How It Works 94
Limitations 95
Licensing 95
Configuring ADC over Gx 95
Monitoring and Troubleshooting 96
Monitor Protocol 96
Show Command(s) and/or Outputs 96
On C-Plane 96
On U-Plane 96

CHAPTER 8 Addition of IP Pool in IP Group 99
Revision History 99
Feature Description 99
How it Works 100
Monitoring and Troubleshooting 100
Show Command(s) and/or Outputs 100
show ip user-plane verbose 100

CHAPTER 9 APN ACL Support 101
Revision History 101
Feature Description 101
Troubleshooting 102

CHAPTER 10 APN AMBR Traffic Policing 103
Revision History 103
Feature Description 103
Limitations 103
Configuring the APN AMBR Traffic Policing Feature 104
Monitoring and Troubleshooting 104
Show Commands and or Outputs 104

CHAPTER 11 APN Data Tunnel MTU Size Configuration 107
Revision History 107
Feature Description 107
Limitation 108
Configuring MTU 108

CHAPTER 12 App-based Tethering Detection in User Plane 111
Revision History 111
Feature Description 111
Limitation 112
Configuring App-based Tethering Detection 112
Enabling App-based Tethering Detection at Rulebase Level 112
Enabling App-based Tethering Detection at Ruledef Level 112
Monitoring and Troubleshooting the App-based Tethering Detection 113
Show Commands and Outputs 113

CHAPTER 13 APN and APN Profile-Based User Plane Selection for CUPS 115
Revision History 115
Feature Description 115

How It Works 115
Architecture 117
Session Recovery and ICSR 117
Limitations 117
Licensing 117
Configuring APN-Based UP Grouping 118
Configuring User Plane Group in Control Plane 118
Configuring User Plane Group 118
Configuring Peer Node ID and User Plane Node IP Address 118
Verifying the User Plane Group 118
Associating User Plane Group with APN 119
Configuring User Plane Group in APN 119
Verifying the User Plane Group in APN 119
Method of Procedure (MOP) to Remove or Change User Plane Group from APN 119
Associating User Plane Group with APN Profile 120
Configuring User Plane Group in APN Profile 120
Monitoring and Troubleshooting APN-Based UP Grouping 120

CHAPTER 14 **Cisco Ultra Traffic Optimization with VPP 121**
Revision History 121
Feature Description 121
RCM Support 122
Sending the GBR or MBR Values to Cisco Ultra Traffic Optimization 122
Cisco Ultra Traffic Optimization Library Deinitialization 123
How it Works 123
Architecture 123
Limitations 124
Show Commands and Outputs 124
Show Commands and Outputs 124
Bulkstats 126
Sample Configuration 130

CHAPTER 15 **Charging Action Configuration Change Support for Existing Sessions Gy and Gz Interface 133**
Revision History 133

Feature Description 133
How It Works 134
Configuration Change Under Charging Action from Rated to Free and Free to Rated Mid Call 134
Configuration Change for Addition of High Priority Rule with Different Charging Action with Different Rating Group 134
Configuration Change for Charging-Action Mid Call for the Ruledef 134
URR Bucket Checkpointing Enhancement for Gy 134

CHAPTER 16 **Dedicated Bearer Establishment without PCRF** 135
Revision History 135
Feature Description 135
How it Works 135
Sx Interface Changes 137
Trigger Action Report IE (Private IE) 138
Trigger Actions 138
N-1 Compatibility Matrix 139
Configuring active-charging-services 139

CHAPTER 17 **Disable Radius Accounting** 141
Revision History 141
Feature Description 141
Configuring RADIUS Accounting on Dedicated Bearer Feature 142
Enabling RADIUS Accounting for All Bearers 142
Disabling RADIUS Accounting for a Specific Bearer 142
Enabling RADIUS Accounting only for the Default Bearer 143

CHAPTER 18 **Default and Dedicated Bearer Support for Pure-P and Collapsed Sessions** 145
Revision History 145
Feature Description 145
Supported Functionality 146
Limitations 147

CHAPTER 19 **Device ID in EDNS0 Records** 149
Revision History 149

Feature Description 149
How it Works 150
Process Flow 150
EDNS0 Packet Format 151
EDNS0 with IP Readdressing 152
Behavior and Restrictions 152
Limitation 153
Configuring EDNS Format and Trigger Action 153
Sample Configuration 154
Monitoring and Troubleshooting 155
Show Commands and Outputs 155
Bulk Statistics 157

CHAPTER 20 **DSCP Markings For Collapse Calls** 159
Feature Summary and Revision History 159
Feature Description 159
How It Works 160
Configuration 160
Monitoring and Troubleshooting 161
Show Commands Outputs 161
SMGR CP Changes 161

CHAPTER 21 **Dynamic and ADC Charging Rule Names** 165
Revision History 165
Feature Description 165

CHAPTER 22 **Dynamic APN and IP Pool Support** 167
Revision History 167
Feature Description 167
How It Works 167
Limitations 169
Configuring Dynamic APN and IP Pool Support 169
Updating the APN Configuration 170
Verifying Dynamic APN and IP Pool Support 170

CHAPTER 23 **Dynamic User Plane Selection** 173
Revision History 173
Feature Description 173

CHAPTER 24 **ECS Regular Expression Support** 191
Feature Summary and Revision History 191
Feature Description 191
How It Works 192
Configuring Regex Rule 193
Configuring Regex Rule via RCM 194
Configuring Regex Rule via PFD Push 194
Sample Configuration 194
Monitoring and Troubleshooting 194
Show Commands and Outputs 194

CHAPTER 25 **End Marker Packets** 197
Revision History 197
Feature Description 197

CHAPTER 26 **Enterprise Onboarding in CUPS** 199
Feature Revision History 199
Feature Description 199
Operational Use Case 200
Architecture 200
Installation 201
How it Works 201
Pre-Processing 202
CP and UP Configuration 203
Post-Processing 204
Add Operation 206
Modify Operation 207
Delete Operation 207
Password Encryption 207

Onboarding Application – Usage and Input Parameters 209
CUPSinfo.txt 209
ADD_ENTERPRISE_INPUT_PARAMETERS.txt 211
MODIFY_ENTERPRISE_INPUT_PARAMETERS.txt 214
DELETE_ENTERPRISE_INPUT_PARAMETERS.txt 215
System Limits 216
Enterprise Onboarding in CUPS OAM Support 218
Show Commands 218
show cups-resource session summary 218
show ip user-plane verbose 218
Error Codes 218

CHAPTER 27 **Event-based CDRs for CUPS 221**
Revision History 221
Event-based CDRs for CUPS 221
Feature Description 221
How It Works 222
Fetching the Usage Report 222
Tariff Time 223
Event Trigger 223
Standards Compliance 224
Monitoring and Troubleshooting 224
Show Commands and/or Outputs 224
show active-charging subscribers full callid call_id urr-info 224
show subscribers user-plane-only callid call_id urr full all 224

CHAPTER 28 **Event Data Records in CUPS 225**
Revision History 225
Feature Description 225
TCP Fast Open 226
How It Works 226
Limitations 229
Configuring Event Data Records in CUPS 229
Configuration on CP to Push EDRs to UP 229

Configuration to Enable EDR Module on UP 230
Configuring Additional TCP Fields 230
Monitoring and Troubleshooting 230
show user-plane-service statistics rulebase name rulebase_name 230
show active-charging rulebase statistics real-time 231
show active-charging edr-format all 232
Bulks Statistics 233

CHAPTER 29 **Error Indication and GTPU Path Failure Detection** 235
Revision History 235
Feature Description 235
How It Works 236
Error Indication Support 236
Error Indication Handling at CP 236
Error Indication Handling on UP 236
Error Indication Generation on UP 237
Error Indication Call Flows 237
GTPU Path Failure Support 240
GTPU Path Failure Support at CP 240
GTPU Path Failure Support at UP 241
Limitations 242
Configuring Error Indication and GTPU Path Failure on Control Plane 242
Configuring Error Indication on CP 242
Configuring GTPU Path Failure on CP 243
Limitations 244

CHAPTER 30 **Firewall Support in CUPS** 245
Revision History 245
Feature Description 245
Overview 246
Configuring the Default Firewall Feature 246
Enabling Firewall for IPv4 and IPv6 247
Configuration Support for Subscriber Firewall 247
Monitoring and Troubleshooting 248

Show CLIs for CUPS 249
SNMP Traps 249
Reassembly Behavior Change 250

CHAPTER 31 GTPC Peer Record and Statistic Optimization 251
Revision History 251
Feature Description 251
How it Works 251
Limitations and Restrictions 252
Configuring the Peer Salvation Functionality 252
gtpc peer-salvation (context configuration mode) 253
gtpc peer-salvation (eGTP service configuration mode) 253
Monitoring and Troubleshooting 254
show egtp-service all 254
show session subsystem debug-info 254
show demux-mgr statistics egtpinmgr all 254
show demux-mgr statistics egtpegmgr all 254

CHAPTER 32 Gx-alias Enhancement 257
Revision History 257
Feature Description 257
How it Works 257
Call Flow 258
Limitation 260

CHAPTER 33 Gy Multiple MSCC and FUI-Redirection 261
Revision History 261
Feature Description 261
Limitations 262

CHAPTER 34 ICSR Bulk Statistics 263
Revision History 263
Feature Description 263
Configuring the ICSR Bulk statistics Schema 263

Show CLIs 264
Bulk Statistics 264

CHAPTER 35 Idle Timer for SAE-GW Sessions 267
Revision History 267
Feature Description 267
Limitations 267
Configuring Idle Timer for SAE-GW Sessions 268

CHAPTER 36 IFTASK Hyperthreading 269
Revision History 269
Feature Description 269
How it Works 269
Limitations and Restrictions 270
Configuring CPU Isolation 270

CHAPTER 37 Indirect Forwarding Tunnel 271
Revision History 271
Feature Description 271
How It Works 272
Call Flow 272
Supported Functionality 275
Configuring Indirect Forwarding Tunnel 275
Enabling Indirect Forwarding Tunnel Feature 275
Verifying the Indirect Forwarding Tunnel Feature 276
show sgw-service name <service_name> 276
Monitoring and Troubleshooting 276
Show Commands Input and/or Outputs 276
show subscribers saegw-only full all 276
show subscribers user-plane-only callid <call-id> pdr all 276
show subscribers user-plane-only full all 277

CHAPTER 38 IP Pool Management 279
Revision History 279

Feature Description 279
How It Works 280
Handling UP De-Registration 280
Hold Timer 280
IP Pools per Context 282
IP Resource Management 282
IP Resource Replenishment/Withdrawal Procedure 282
No-chunk-pool for One UP per UP Group 283
Static IP Pool Management 284
UP Selection 284
UP Selection based on IP Pool Chunk Availability 285
Supported Functionality 285
Limitations 285
Configuring IP Pool Management 286
At Control Plane 286
Configuring Chunk-size Value 288
At User Plane 288
Configuring User Planes for a System 288
Monitoring and Troubleshooting 289
Show Command(s) and/or Outputs 289
show ip pool-chunks pool-name <pool-name> 289
show ip pool-chunks pool all 290
show ip pool-chunks up-id <up_id> user-plane-group name <grp-name> 290
show ip user-plane chunks 291
show ip user-plane prefixes 291
show ip user-plane verbose 291
show ip user-plane 293
show ipv6 pool-chunks pool-name <pool-name> 293
show ipv6 pool-chunks up-id <up_id> user-plane-group name <grp-name> 293

CHAPTER 39 **IP Source Violation** 295
Revision History 295
Feature Description 295
Configuring IP Source Violation 295

Monitoring and Troubleshooting 296
Show Command(s) and/or Outputs 296
show sub user-plane-only full all 297

CHAPTER 40 IPSec in CUPS 299
Revision History 299
Feature Description 299
IPSec AH and ESP 299
IPSec Transport and Tunnel Mode 300
IPSec Terminology 300
Crypto Access Control List 300
Transform Set 300
ISAKMP Policy 300
Crypto Map 301
Crypto Template 301
DSCP Marking of ESP Packets 301
Application Configured with DSCP Value 301
Crypto Map Configured with DSCP Value 302
Application and Crypto Map Configured with DSCP Value 303
Supported Algorithms 304
Limitations and Restrictions 305
Configuring DSCP in Crypto Map 306
Sample Configuration 306
Configuring QoS 307
Monitoring and Troubleshooting 308
Show Commands and Outputs 308

CHAPTER 41 L2 Marking Support 313
Revision History 313
Feature Description 313
How it Works 313
Limitations 315
Configuring L2 Marking Support 315
Configuring Internal Priority 315

Associating QCI-QoS Mapping Table 316
Configuring QCI Derived L2 Marking 316
Associating L2 Mapping Table 317
Configuring DSCP Derived L2 Marking 317

CHAPTER 42 L3, L4, and L7 Rule Combination in Ruledef 319
Revision History 319
Feature Description 319
How it Works 320
Configuring the L3, L4, and L7 Rule Combination in Ruledef Feature 320
Verifying the L3, L4, and L7 Rule Combination in Ruledef Feature Configuration 320
Monitoring and Troubleshooting 321
Show commands and Outputs 321

CHAPTER 43 L7 PCC Rules 323
Revision History 323
Feature Description 323
How It Works 324
Content Filtering 324
DNS 326
DNS Snooping 326
FTP 327
HTTP 328
HTTPS 330
HTTP URL Filtering 330
RTP/RTSP 333
RTP Dynamic Flow Detection 334
Rule-matching for Bearer-specific Filters 334
SIP 335

CHAPTER 44 Local Policy in CUPS 337
Revision History 337
Feature Description 337
How It Works 338

Configuring Local Policy in CUPS 338

CHAPTER 45 **Load/Overload and UP Data Throttling Support on Sx 341**

Feature Description 341

How It Works 341

User Plane Selection 341

Node-level Load/Overload Support 342

Sx Establishment Request Throttling at CP in Overload State 342

Sx Establishment Request Throttling at UP in Self-Protection 342

Session Termination Trigger from UP in Self-Protection 342

Limitation 343

Configuring Load and Overload Support 343

User Plane Load Control Profile Configuration 343

User Plane Overload Control Profile Configuration 344

Associating a Load Control Profile with a User Plane Service 347

Sx Protocol Configuration on Control Plane 347

Monitoring and Troubleshooting 347

Show Commands Input and/or Outputs 347

show userplane-load-control-profile name name 347

show userplane-overload-control-profile name name 348

show user-plane-service statistics all 349

show sx service statistics all 350

Bulk Statistics 350

SNMP Traps 351

CHAPTER 46 **LTE - Wi-Fi Seamless Handover in CUPS 353**

Revision History 353

Feature Description 353

How It Works 354

LTE - Wi-Fi Handover 354

ICSR and Session Recovery 355

Limitations 355

Standards Compliance 355

Configuring LTE and Wi-Fi Seamless Handover 355

Monitoring and Troubleshooting 356
Show Command(s) and/or Outputs 356
show apn statistics name <name> 356

CHAPTER 47 **Monitor Subscriber for CUPS** 357
Revision History 357
Feature Description 357
Monitor Subscriber Sx Private IE 359
Control Plane SMGR Functionality 363
User Plane SMGR Functionality 363
Multi PDN Multi Trace 364
MonSub Stats 365
X-Header 365
How It Works 365
Configuration Procedure for Monitor Subscriber 365
Monsub CLI Options 366
Context, CDRMOD and Hexdump Interaction for Monitor Subscriber 368
PCAP File Name Convention 368
PCAP File Location 370
Limitations 371
Configuring the Hexdump Module for MonSub in UPF 372
Configuring MonSub Poll Timer 372
Configuring MonSub File Name 373
Monitoring and Troubleshooting 374
SNMP Traps 374

CHAPTER 48 **MPLS Support on VPC-SI for CUPS** 375
Revision History 375
Feature Description 375
How it Works 376
MPLS-CE Connected to PE 376
VPC-SI as a PE 377
Overview 377
Sample Configuration 377

IPv6 Support for BGP MPLS VPNs 378
Overview 378
Sample Configuration 379
VPN-Related CLI Commands 382
Monitoring and Troubleshooting 387
Show Commands and Outputs 387
show mpls fn vpp 387

CHAPTER 49 Multiple Control Plane Support on User Plane 389
Revision History 389
Feature Description 389
How it Works 390
Configuring Multiple Control Plane Support on User Plane 392
Disabling PFD Configuration Push from CP 392
Configuring Multiple CP on UP 392
Monitoring and Troubleshooting 393
Show Commands and/or Outputs 393
show sx-service statistics address <ip_address> 393
show user-plane-service statistics peer-address <ip_address> 395
show ip chunks peer <ip_address> 397
show ipv6 chunks peer <ip_address> 397
Sample RCM Configuration 398

CHAPTER 50 Multiple UP Group Support 405
Revision History 405
Feature Description 405
Relationships 405
Architecture 406
Components 406
How It Works 407
Limitations and Restrictions 407
Configure the Multiple UP Group Support Feature 407

CHAPTER 51 MOCN Special Handling of CRA and CNR 409

Revision History 409
Feature Description 409
TAI Change Event Handling 410
How It Works 411
Start Reporting TAI Change 411
Stop Reporting TAI Change 412

CHAPTER 52 N+2 UP Recovery 415
Revision History 415
Revision History 415
Feature Description 415
Deployment Architecture 416
Limitations 417
How It Works 417
Call Flows 418
SAEGW Detach and Reattach on Path Failure 418
P-GW Detach and Reattach on Path Failure 420
S-GW Detach and Reattach on Path Failure 422
GnGp GGSN Detach and Reattach on Path Failure 424
Additional N+2 Handling Scenarios 426
Double Failure Handling Scenarios 430
BFD Flapping and VPC 430
Sx-association Scenarios 431
N+2 and IP Addressing 432
Loopback IP Addresses 432
IP Address Availability 432
Configuring N+2 UP Recovery 432
Monitoring and Troubleshooting 434
Show Commands 434
SNMP 434

CHAPTER 53 NAT Support 437
Feature Summary and Revision History 437
Revision History 437

Feature Description 437
Limitations 438
Configuring NAT in CUPS 439
Sample Configurations 439
Control Plane 439
User Plane 439
Monitoring and Troubleshooting 440
Gathering NAT Statistics in CUPS 440
Clear Commands 441
SNMP Traps for NAT Parameter Thresholds 441
Bulk Statistics 442
Context Schema 442
ECS Schema 443
NAT-realm Schema 444
EDRs 446
Sample EDR 446
NAT Binding Records 446
Sample NBR 447
Packet Drop EDR 447
Sample Packet Drop EDR 447

CHAPTER 54 NAT ALG Support 449
Feature Summary and Revision History 449
Revision History 449
Feature Description 449
Components of Session Initiation Protocol ALG 450
How it Works 452
FTP 453
RTSP 453
PPTP 453
SIP 453
TFTP 453
H323 454
NAT FW Processing 454

Uplink Packet Processing 455
Downlink Packet Processing 455
Configuring NAT ALG 455
Sample Configuration for FTP NAT ALG 456
Sample Configuration for RTSP NAT ALG 457
Sample Configuration for PPTP NAT ALG 457
Sample Configuration for TFTP NAT ALG 458
Sample Configuration for H323 NAT ALG 459
Sample Configuration for SIP NAT ALG 459
Monitoring and Troubleshooting 460

CHAPTER 55 N : M Redundancy 465
Revision History 465
Feature Description 465
Configuring Ignore SSH IP Installation 466

CHAPTER 56 Netloc and RAN/NAS Cause Code 467
Revision History 467
Feature Description 467
Configuring Netloc and RAN/NAS Cause Code 468

CHAPTER 57 Network Provided Location Indication 469
Revision History 469
Feature Description 469
How It Works 469
Supported Functionality 470
Limitations 470

CHAPTER 58 Nexthop Forwarding Support IPv4/v6 Address 471
Revision History 471
Feature Description 471
How It Works 471
Architecture 471
Configuring Nexthop Forwarding Support IPv4/IPv6 Address 475

Configuring Nexthop Forwarding at APN Configuration Mode 475
Configuring Nexthop Forwarding at IP Pool 475
Configuring Nexthop Forwarding Through AAA 476
Monitoring and Troubleshooting 476
Show Commands and Outputs 476

CHAPTER 59 **Network Trigerred Service Restoration** 477
Feature Description 477
Configuring NTSR 477
APN Profile Configuration 478
Peer Profile Configuration (Ingress) 478
NTSR Pool Configuration 478
S-GW Service Access Peer Map Association 479
Monitoring and Troubleshooting 479
Show Commands Input and/or Outputs 479
show apn-profile full all 479
show apn-profile full name apn_name 479
show ntsr-pool all 480
show ntsr-pool full all 480
show ntsr-pool full pool-id pool_id 480
show ntsr-pool pool-id pool_id 480
show sgw-service statistics all 480
show subscribers sgw-only full all 481

CHAPTER 60 **NSO-based Configuration Management** 483
Revision History 483
Feature Description 483
Use Cases 483
How it Works 484
Architecture 484
RCM and NSO 485
Components 485
Minimum Platform, Hardware, and Software Requirements 486
Licensing 486

NSO Installation 486
Call Flows 487
Onboarding Existing 4G CUPS VNFs into NSO 487
4G CUPS Device Configuration Push – Manual 488
Configuration Push from NSO to 4G CUPS UPs in N:M Redundancy – Automated 489
Configuration Metadata Pre-population 490
NSO HA Switchover Handling 491
Recovery 491
CP Switchover (1:1) 492
UP Switchover (1:1) 492
UP Switchover (N:M) 492
Out-of-Band Configuration 493
Sensitive Elements in Configuration 493
Lawful Intercept 493
CUPS Configuration MOP 494
Device Onboarding 494
RESTCONF 494
CLI 495
Prepopulating Configuration Metadata 495
RESTCONF 498
CLI 499
Configuration Push through Mobility MOP 499
Configuration MOP Push Request Flow 499
Configuration MOP Rollback Request Flow 500
MOP Automation 501
Configuration Prerequisites 501
Mop-type Pair Prerequisites 502
NSO APIs 502
UP Configuration Push and Recovery in N:M Redundancy 516
NETCONF Notification Subscription on NSO 517
Handle RCM UP Recovery Notification 517
RCM UP Config-Push Notification 518
UP Day-0.5 Update 520
Prerequisites for Configuration Push 521

Limitations and Restrictions 522
Troubleshooting 523
Appendix A: Incompatible StarOS Native Command Syntax 524
Appendix B: Example Configurations for N:M Deployment with RCM 527
Host-specific Configuration-UP 527
First Active UP 527
Second Active UP 528
Host-specific Configuration-RCM 529
First Active RCM 529
Second Active RCM 531
Common Configuration 533
Standby Configuration (Active1 + Active2) 536

CHAPTER 61 **NSO Orchestration for 4G CUPS** 539
Revision History 539
Feature Description 539
Use Cases 539
How it Works 540
Architecture 540
Minimum Platform and Software Requirements 543
Network and Hardware Requirements 544
Licensing 544
Call Flows 544
VNF Onboarding 545
P2P Module Installation 546
VNF Termination 546
Recovery 546
Limitation 547
Installing NSO Packages 547
VNF Orchestration/Deployment and Automatic Configuration Management 548
Pre-population of Config Metadata for VNF Orchestration 548
Onboarding ESC and Openstack as Devices 553
Prerequisites for VNF Instantiation 558
VNF Instantiation 559

VNF Instantiation - Component Interactions and Flows 564
Checking the VNF Instantiation Status 566
VNF Dashboard 567
VNF Deletion 567
Checking the VNF Deletion Status 568
Removing Configuration Metadata 568
Cleaning Config Files from NSO Filesystem 568
Automation Process - VNF Deployment, Onboarding, and Configuration Push 568
Instantiation of VNF using Input Payload 568
Onboarding VNF as a Device in NSO 568
Upgrading the P2P Module in VPC Device 568
Configuration Push to the Onboarded Device 568
Appendix A: YANG definition of VNF 569

CHAPTER 62 **NSH Traffic Steering** 577
Revision History 577
Feature Description 577
Post Processing Rule Condition Match for Traffic Steering 578
BFD Instance Id Configuration in UP Appliance Group Using Interface Names 578
Architecture—Standalone Mode 578
Components 580
Limitations 581
How it Works—Standalone Mode 582
Packet Flows 582
NSH Traffic Steering Requirements 583
SFP Selection 585
Interworking with Inline Features 585
Configuring the L2 and NSH Traffic Steering Feature—Standalone Mode 586
N to M Traffic Steering 590
Monitoring and Troubleshooting—Standalone Mode 595
SNMP Traps 601
Bulk Statistics 601
Feature Description—Sandwich Mode 602
Architecture—Sandwich Mode 602

How it Works—Sandwich Mode 604
Packet Flows in Sandwich Mode 604
Service-Scheme Selection for Traffic Steering 607
Default Service Chain 608
SFP Selection 608
Limitations and Restrictions 608
Configuring NSH Traffic Steering—Sandwich Mode 609
CP Configuration 609
UP Configuration 610
Configuring Post Processing Ruledef in Both Standalone and Sandwich Mode 612
Configuring BFD Instance Id Using Interface Name in UP Appliance Group 612
Monitoring and Troubleshooting the NSH Traffic Steering—Sandwich Mode 613
Show Commands 613
show user-plane traffic-steering up-appliance-group all 614

CHAPTER 63 **Packet Flow Description Management Procedure for Static and Predefined Rules** 615
Feature Description 615
How It Works 615
Moving Bulk Configurations from Control Plane to User Plane 616
Limitation 618
Sx Association 618
Configuring Control Plane Group 620
Monitoring and Troubleshooting Sx Association 623
Monitoring and Troubleshooting 626
Show Command(s) and/or Outputs 626
show user-plane-service charging-action all 626
show user-plane-service charging-action name charging-action-name 628
show user-plane-service rule-base all 629
show user-plane-service rule-base name rule-base-name 631
show user-plane-service rule-def all 633
show user-plane-service rule-def name rule-def-name 634

CHAPTER 64 **PDI Optimization** 635
Feature Summary and Revision History 635

Revision History 635
Feature Description 635
Relationships 636
How It Works 636
PDI Optimization Changes on Control Plane 636
Create Traffic Endpoint IE 637
Created Traffic Endpoint IE 638
Update Traffic Endpoint IE 638
Remove Traffic Endpoint IE 639
PDI Changes in Create PDR 639
PDI Optimization Changes on User Plane 639
Handling of Create Traffic Endpoint 639
Handling of Update Traffic Endpoint 639
Handling of Remove Traffic Endpoint 640
Handling of Create PDR 640
Session Recovery and ICSR 640
Control Plane 640
User Plane 641
Standards Compliance 641
Limitations 641
Configuring the PDI Optimization Feature 641
Enabling PDI Optimization 641
Verifying the PDI Optimization Feature Configuration 642
PDI Optimization OAM Support 642
Show Command Support 642
show subscribers user-plane-only callid <call_id> pdr all 642
show subscribers user-plane-only callid <call_id> pdr full all 642

CHAPTER 65 **P-GW CDR in CUPS** 643
Revision History 643
Feature Description 643
Limitations 644
User Location Information in P-GW CDR 644

CHAPTER 66 P-GW Restart Notification 647
Revision History 647
Feature Description 647

CHAPTER 67 Post Processing Interaction for DCCA 649
Feature Description 649
Normal Rule Matching 649
Application Processing 650
Post Processing 650
Limit Reached Post Processing 651
Configuring Post Processing 651

CHAPTER 68 Priority Recovery Support for VoLTE Calls 653
Feature Summary and Revision History 653
Feature Description 653
How It Works 653
Call Flows 655
Configuration 656
Monitoring and Troubleshooting 657
Show Commands and Outputs 657

CHAPTER 69 QoS Group of Ruledefs Support 659
Revision History 659
Feature Descriptions 659
How It Works 659
Data Path Enforcement 660
Static Configuration Push to UPlane 660
QGR Params Push to UPlane 660
Processing of QGR on UPlane 661
QGR Hit in Data Path 662
Limitations 662
Monitoring and Troubleshooting 662
Show Commands and Outputs 662

CHAPTER 70 Rate Limiting Function (RLF) 669
Revision History 669
Feature Description 669

CHAPTER 71 S2a Interface Support 671
Revision History 671
Feature Description 671

CHAPTER 72 S2b Interface Support 673
Feature Description 673

CHAPTER 73 S-GW CDR in CUPS 675
Revision History 675
Feature Description 675

CHAPTER 74 S-GW New Call Rejection 677
Feature Description 677
How It Works 677
Limitations 678
Configuring S-GW New Call Rejection 678
Enabling New Call Rejection 678
Monitoring and Troubleshooting 679
Show Command(s) and/or Outputs 679
show saegw-service statistics all function sgw 679
show sgw-service name 679

CHAPTER 75 S-GW Session Idle Timeout 681
Revision History 681
Feature Description 681
Configuring Session Idle Timeout 682

CHAPTER 76 SAEGW Idle Buffering with DDN Delay and DDN Throttling 683

Revision History 683
Feature Description 683
How It Works 683
Downlink Data Notification – Delay (DDN-D) Support 684
DDN Throttling Support 685
No User Connect Timer Support 685
DDN Call Flows 686
DDN Success Scenario 686
DDN Failure Scenario 687
No User Connect Timer Support 688
DDN Delay Timer 689
Sx Interface 690
Limitations 692
SAEGW Idle Buffering with DDN Delay and DDN Throttling Support Configuration 693
DDN Throttling for Release 10 Compliant MME 693
DDN Throttling for non-Release 10 Compliant MME 693
Show Commands Input and/or Outputs 695
show subscribers user-plane-only-full all 695

CHAPTER 77 **Secondary RAT Usage Report in CDR Records** 697
Revision History 697
Feature Description 697
Behavior Matrix 698
Relationship to Other Features 700
Limitations 700
Configuring Secondary RAT Usage Report through GTPP 701
Enabling or Disabling the Secondary RAT Usage Report 701
Controlling the Maximum Number of Entries 701
Suppressing Zero-Volume Secondary RAT Usage Report 704
Monitoring and Troubleshooting 704
Show Commands and Outputs 704
show config 704
show config verbose 705
show gtpp group 705

show gtpp statistics group 705

CHAPTER 78 Self-overload Detection and Admission Control of Sx at UP 707

Revision History 707

Feature Description 707

Limitations 708

Configuring Overload Control at User Plane 708

eMPS Profile Creation and Association to S-GW and P-GW Services of Control Plane 708

Configuring the Overload Control Profile at UP 709

Configuring Overload Threshold Parameters 709

Configuring System Weightage Parameters 709

Configuring Session Manager Weightage Parameters 710

Associating an Overload Control Profile with a User Plane Service 710

Monitoring and Troubleshooting 710

Show Commands Input and/or Outputs 710

show user-plane-service name name 710

show user-plane-service statistics name user_plane_service_name 711

show userplane-overload-control-profile name name 711

CHAPTER 79 Smart Licensing 713

Revision History 713

Overview 713

Cisco Smart Software Manager 714

Smart Accounts/Virtual Accounts 714

Smart Licensing Mode 714

Request a Cisco Smart Account 715

Software Tags and Entitlement Tags 715

Configuring Smart Licensing 718

Monitoring and Troubleshooting Smart Licensing 719

CHAPTER 80 Software Management Operations 721

Revision History 721

Overview 721

SNMP Traps 722

Limitations 723
Upgrading or Downgrading of CP and UP 723
Health Checks 723
Build Upgrade 725
CP Upgrade 726
UP Upgrade 727
CP and UP Upgrade 727
Downgrade Procedure 728

CHAPTER 81 Standard QCI Support 731
Revision History 731
Feature Description 731
Limitations 731

CHAPTER 82 Static and Predefined Rule Match Support for Shallow Packet Inspection 733
Revision History 733
Feature Description 733
How It Works 734
Monitoring and Troubleshooting 734
Show Command(s) and/or Outputs 735
show subscribers user-plane-only full all 735
show subscribers user-plane-only callid <callid> pdr full all 735
show subscribers user-plane-only seid <seid> pdr full all 735
show subscribers user-plane-only callid <callid> pdr id <id> 735
show subscribers user-plane-only seid <seid> pdr id <id> 736

CHAPTER 83 Static IP Assignment from RADIUS 737
Feature Description 737
How it Works 737
Limitations 737

CHAPTER 84 Suspend and Resume Notification for Pure-S Calls 739
Revision History 739
Feature Description 739

How It Works 739
Call Flows 740
Suspend Notification 740
Resume Notification 740

CHAPTER 85 TACACS+ Over IPSec 743
Revision History 743
Feature Description 743
Architecture 743
Deployment Architecture 744
How it Works 745
Encryption of TACACS+ Client Data 745
Decryption of TACACS+ Server Data 746
Recovery 748
Limitation 748
Configuring TACACS+ over IPSec 748
Configuring TACACS+ Configuration Mode 749
Provisioning TACACS+ with IPSec 749
Provisioning TACACS+ with IPSec in Tunnel Mode 749
Provisioning TACACS+ with IPSec in Transport Mode 750
Monitoring and Troubleshooting 751
Show Commands and Outputs 751

CHAPTER 86 Tariff Time Support 753
Revision History 753
Feature Description 753

CHAPTER 87 UP Call Summary Log 755
Revision History 755
Feature Description 755
How it Works 756
Fault and Fault Reporting 758
Redundancy 758
Interdependencies 758

Limitations and Restrictions 759
Configuring Call Summary Log in UP 759
Enabling/Disabling the CSL 759
UP Service Configuration 759
Monitoring and Troubleshooting 760
Statistics 760
Show Command Outputs 760

CHAPTER 88 URL Blockedlisting 761
Revision History 761
Feature Description 761
How it Works 761
Limitations 762
Configuring URL Blockedlisting 763
Loading URL Blockedlisting Database on UP 763
Configuration to Enable URL Blockedlisting 763
URL Blockedlisting Database Upgrade 764
Monitoring and Troubleshooting 764
Show Command(s) and/or Outputs 764
show user-plane-service url-blacklisting database 764
show user-plane-service url-blacklisting database url database_directory_path 765
show user-plane-service url-blacklisting database facility sessmgr all 765
show user-plane-service inline-services info 766
show user-plane-service rulebase name rulebase_name 766
show user-plane-service inline-services url-blockedlisting statistics 766
show user-plane-service inline-services url-blacklisting statistics rulebase name rulebase_name 766
Bulk Statistics 766
SNMP Traps 766

CHAPTER 89 User Plane Selection based on TAC Range 769
Revision History 769
Feature Description 769
How It Works 769
Limitations 770

Configuring User Plane Selection based on TAC Range 771
Configuring Tracking Area Code Range 771
Verifying the Tracking Area Code Range Configuration 772
Configuring Tracking Area Code Profile 772
Verifying the Tracking Area Code Profile Configuration 772
Configuring Routing Area Code Profile 773
Verifying the Routing Area Code Profile Configuration 773

CHAPTER 90 User Plane Node Bring Down Procedure 775
Revision History 775
Feature Description 775
Preconditions 776
How it Works 776
Call Flow 776
UP Selection when a UP is Marked Busy Out 776
UP Clear Idle Subscribers based on Busy Out Inactivity Timeout 777
Limitations and Considerations 777
Configuring UP Node Bring Down Procedure 778
Monitoring and Troubleshooting 778
Show Commands and Outputs 778
show sx peers 778
show sx peers wide 779

CHAPTER 91 Virtual APN in CUPS 781
Revision History 781
Feature Description 781
How It Works 782
Call Flow 782
Limitations 783
Configuring Virtual APN in CUPS 784

CHAPTER 92 VoLTE Support in CUPS 787
Revision History 787
Feature Description 787

How It Works 788
Call Flows VoLTE Support 788
Handling Suspend Notifications 788
Handling Resume Notifications 789
Limitations 790

CHAPTER 93 **Volume Reporting over Gx** 791
Revision History 791
Feature Description 791
How it Works 792
Control Plane Handling for VoGx 792
User Plane Handling for VoGx 793
Limitations 793
Configuring VoGx Monitoring Key Range 793
Monitoring and Troubleshooting VoGx 794
Show Commands and/or Outputs 794

CHAPTER 94 **VPN Manager Recovery Support** 797
Feature Summary and Revision History 797
Feature Description 797

CHAPTER 95 **VPP Support** 799
Revision History 799
Charging Support 800
Delay-Charging Via Rule Base 800
Flow Idle-time Out 801
HTTP Support 801
IP Readdressing 801
DNS Readdress Server List 801
LTE Handover 803
Next Hop 803
PDN Update 803
Policing 803
Pure-S Support 804

Response-based Charging via Service Schema 805
Response-based TRM via Service Schema 805
ToS Marking 805
Volume-based Offload 805
Supported Functionality 805
Limitations 806
Enabling Fast Path in User Plane Service 807
Enabling VPP on SI Platform 807
Monitoring and Troubleshooting VPP Fast Path 807
Support for VPP Configuration Parameters Override 808

CHAPTER 96 **VRF Support for CUPS** 809
Revision History 809
Feature Description 809
VPNMgr Crash Outage Improvement for IP Pool under VRF 810
Configuring VRF 811
Monitoring and Troubleshooting 813
Show Command(s) and/or Outputs 814
show ip chunks 814
show ipv6 chunks 814

CHAPTER 97 **X-Header Insertion and Encryption** 815
Revision History 815
Feature Description 815
How It Works 815
X-Header Insertion 815
X-Header Encryption 816
Configuring X-Header Insertion and Encryption 816
Configuring X-Header Insertion 817
Configuring X-Header Encryption 818
Verifying the X-Header Insertion and Encryption Configuration 819
Monitoring and Troubleshooting the X-Header Insertion and Encryption feature 819

APPENDIX A **IP Pool Planning Guidelines** 821

IP Distribution in CUPS Architecture 821
UP Group Concept 821
Default UP Group 822
Specific UP Group 822
When to Add New Pool 822
IP Pool Fine-tuning Parameters 824
Threshold Timer 824
Chunk Withdrawal 824
Initial Chunk Pushed 824
Chunk Size 825
Dynamic IP Pool Planning Guidelines 825
Chunking Guidelines 825
UP Grouping Guidelines 826
UP Addition Guidelines 826
Miscellaneous Guidelines 826
Static IP Pool Guidelines 827

About this Guide

Note Control and User Plane Separation (CUPS) represents a significant architectural change in the way StarOS-based products are deployed in the 3G, 4G, and 5G networks. This document provides information on the features and functionality specifically supported by this 3G/4G CUPS product deployed in a 3G/4G network. It should not be assumed that features and functionality that have been previously supported in legacy or non-CUPS products are supported by this product. References to any legacy or non-CUPS products or features are for informational purposes only. Furthermore, it should not be assumed that any constructs (including, but not limited to, commands, statistics, attributes, MIB objects, alarms, logs, services) referenced in this document imply functional parity with legacy or non-CUPS products. Please contact your Cisco Account or Support representative for any questions about parity between this product and any legacy or non-CUPS products.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This guide describes the User Plane (UP) functionality in Control and User Plane Separation (CUPS). This document also contains feature descriptions, configuration procedures, and monitoring and troubleshooting information.

- Conventions Used, on page xliii

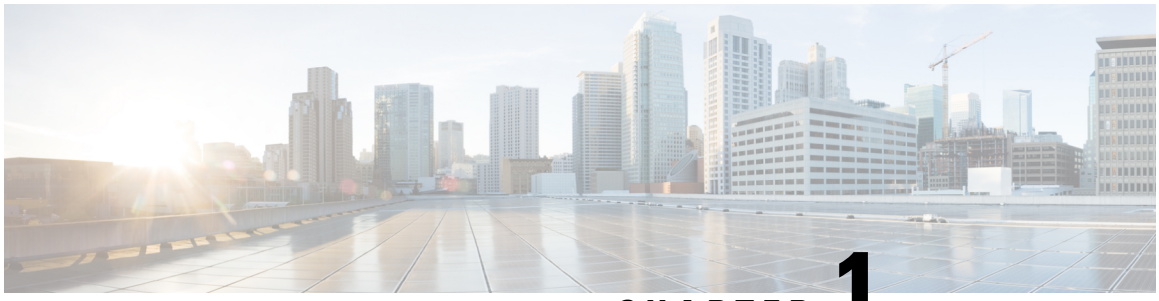
Conventions Used

The following tables describe the conventions used throughout this documentation.

Notice Type	Description
Information Note	Provides information about important features or instructions.
Caution	Alerts you of potential damage to a program, device, or system.

Notice Type	Description
Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **command** *variable*	This typeface represents a variable that is part of a command, for example: **show card** *slot_number* *slot_number* is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**



CHAPTER 1

Overview

The Evolved Packet Core (EPC) network is evolving and moving toward Control User Plane Separation (CUPS) based architecture where User Plane and Control Plane are separate nodes for P-GW, S-GW, and TDF products. The User Plane and Control Plane combined together provide functionality of a node for other elements in the EPC network. However, keeping it separate has numerous advantages from the network point of view – support different scaling for Control Plane and User Plane, support more capacity on per session level in User Plane, and so on.

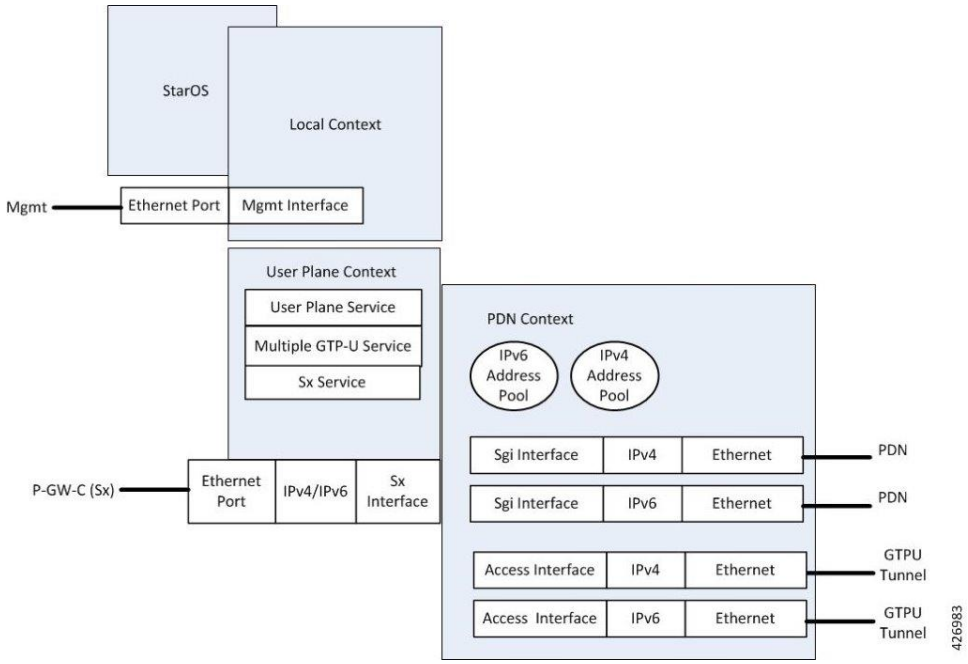
This chapter highlights high-level details, call flows, and configurations related to Control Plane implementation for P-GW, S-GW, and SAEGW products.

- Product Description, on page 1
- Supported Features and Functionality, on page 3
- How It Works, on page 15

Product Description

The SAEGW-U Virtualized Network Function (VNF) can be hosted in Cisco Ultra Services Platform (USP) on COTS hardware or on ASR 5500/DPC2 chassis. The SAEGW-U can be collocated with SAEGW-C in the same data center or can be located remotely in a different data center.

Following is a high-level architecture of User Plane as a service.



Some important points that describe the User Plane as a service:

- User Plane can be programmed from Control Plane.
- Single User Plane service can serve both SGW-U and P-GW-U type sessions.
- Two or more separate User Plane services can be defined for each node type, SGW-U and PGW-U, respectively.
- A group of SAEGW-Us can be explicitly associated with an APN. If no group is associated, a default group is used which includes all the registered User Planes that are registered to SAEGW-C but are not part of any configured SAEGW-U group.
- User Plane service is associated with Sx service for the Control Plane interface, and GTP-U service for receiving GTP-U packets.

Important Currently, each User Plane Service is associated with only single Sx service to interface with Control Plane.

- User Plane service can be associated with four GTP-U services which can be extended to support SaMOG, GGSN, and ePDG.
- Multiple peers of Control Plane services use single User Plane service.
- To associate the IP pool and its configuration, APN configuration is required.

Important Currently, User Plane supports APN and pool configuration. The IP addresses are allocated from the Control Plane and are validated in the User Plane.

Supported Features and Functionality

3GPP ULI Enhanced Reporting Support

This feature enhancement covers ULI-related gaps in P-GW and GGSN as per 3GPP standards.

S4SGSN reports ULI to the P-GW through S-GW. P-GW determines the changes in the ULI with previously received ULI. If P-GW detects any change and the change request is from the PCRF as an event trigger, then the P-GW reports the ULI to the PCRF.

SGSN reports ULI to the GGSN. GGSN determines the changes in the ULI with previously received ULI. If GGSN detects any change and the change request is from the PCRF as an event trigger, then the GGSN reports the ULI to the PCRF. This feature also supports the detection of the change in RAI received as part of the ULI field at GGSN.

For more information on 3GPP ULI Reporting Support Enhancement, refer the *3GPP ULI Reporting Support Enhanced* section in the *StarOS P-GW Administration Guide*.

AAA Server Group

The AAA Server Group feature is used to create and manage the Diameter/RADIUS server groups within the context or system. The AAA server group facilitates management of group (list) of servers at per subscriber/APN/realm-level for AAA functionality.

Note The AAA Server Group is an existing feature that is supported in non-CUPS architecture. With this release, the feature is qualified in CUPS architecture.

For additional information about CLI configurations related to AAA server group, refer the *AAA Server Group Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

APN Configuration Support

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

The CLI commands **radius-group**, **cc-home behaviour 0x10 profile 2** and **mediation-device** are qualified and validated in the CUPS architecture to support APN configuration.

radius-group

Under this functionality validation the CUPS architecture supports 800 Radius Server Groups each group configured with RADIUS Authentication and Accounting server.

cc-home { behavior *bits* | profile *index* }

Configures the home subscriber charging characteristics (CC) used by the GGSN when those from the SGSN will not be accepted. The values configured in the CLI are taken into precedence by CUPS SAEGW service and populated appropriately in the GTPP CDR records.

NOTES:

- **behavior** *bits*: Specifies the behavior bit for the home subscriber charging characteristic. bits can be configured to any unique bit from 001H to FFFH (0001 to 1111 1111 1111 bin) where the least-significant bit corresponds to B1 and the most-significant bit corresponds to B12.
- **profile** *index*: Specifies the profile index for the home subscriber charging characteristic. index can be configured to any integer value between 0 and 15. Default: 8
- For more information, refer to the **cc-home** command under *APN Configuration Mode Commands* chapter in the *Command Line Interface Reference A-B* document

mediation-device [context-name *context_name*] [delay-GTP-response] [no-early-PDUs] [no interims] +

This command and all associated sub section CLIs are supported in CUPS. This CLI enables use of **mediation device** and all associated configuration that can be used for a given APN by CUPS SAEGW service.

NOTES:

- **context-name** *context_name*: Configures the mediation VPN context for this APN as an alphanumeric string of 1 through 79 characters that is case sensitive. If not specified, the mediation context is the same as the destination context of the subscriber. Default: The subscribers destination context.
- **delay-GTP-response**: When enabled, delays the CPC response until an Accounting Start response is received from the mediation device. Default: Disabled.
- **no-early-pdus**: Specifies that the system delays PDUs from the MS until a response to the GGSN accounting start request is received from the mediation device. The PDUs are queued, not discarded. Default: Disabled.
- **no-interim**: Disables sending interims to the mediation server. Default: Disabled.
- For more information, refer to the **mediation-device** command under *APN Configuration Mode Commands* chapter in the *Command Line Interface Reference A-B* document

Asynchronous Core Transfer Support for egtpinmgr

Asynchronous core transfer support for egtpinmgr has been added in CUPS to optimize outage time during an egtpinmgr restart.

Previously, when the egtpinmgr restarted, the recovery process began only after a core dump file was created and transferred. However, the time taken to transfer the core file was significant. The outage time during an egtpinmgr restart was equal to the egtpinmgr recovery time plus the core file transfer time.

Support for Asynchronous Core Transfer has been added in CUPS to include the egtpinmgr during the recovery process. Now, recovery begins when the egtpinmgr process crashes without waiting for the kernel to complete a core dump file transfer and release its resources. As a result, the outage time during an egtpinmgr restart is equal to the egtpinmgr recovery time only.

With this enhancement, outage time during an egtpinmgr restart is reduced. The outage time consists only of the time required to recover the egtpinmgr. The time taken to create and transfer the core file no longer contributes to the outage time.

Note The Asynchronous Core Transfer Support for egtpinmgr is an existing feature that is supported in non-CUPS architecture. With this release, the feature is qualified in CUPS architecture.

Charging Data Records to HDD

A Charging Data Record (CDR) is a formatted collection of information about a chargeable event. The GTPP accounting CDRs that are generated are sent to an external node for storage. The CDRs are written to files in formats supported by the external node and stored on the hard disk (HDD). From the HDD, CDR files can be pushed or pulled using FTP or SFTP protocols.

Note It is strongly recommended that you do not use the system directories created by StarOS under */hd-raid/records/* for deployment use cases such as backups. If such directories are used, this could impact the normal functioning of the product.

CDR is an existing feature that is supported in the non-CUPS architecture, and qualified in the CUPS architecture. For additional information, see the *HDD Storage* chapter in the *GTPP Interface Administration and Reference*.

GTP-C Path Failure Enhancements and Improved Debugging Tools

In CUPS architecture, enhancements have been added to optimize GTP-C path failure functionality, and to improve the debug capability of the system for GTP-C path failure problems. These features will help Operators and Engineers to debug different aspects of the system that will help in identifying the root cause of GTP-C path failures in the network. These enhancements affect path failure detection via the S5, S8, S2b, and S2a interfaces.

The following enhancements are added in CUPS as part of this feature:

- The node can be configured so that it does not detect a path failure if a low restart counter is received due to incorrect or spurious messages. This prevents call loss. The option to disable path failure due to Echo Request/Response and Control Message Request/Response messages is also available so that call loss is prevented in the event of a false path failure detection.
- More granularity has been added to GTP-C path failure statistics so that the root cause of issues in the network can be diagnosed more quickly.
- A path failure history for the last five path failures per peer is available to assist in debugging path failures in the network.
- Seamless path failure handling is implemented so that call loss is avoided during redundancy events.

Note The GTP-C Path Failure Enhancements and Improved Debugging Tools is an existing feature that is supported in non-CUPS architecture. With this release, the feature is qualified in CUPS architecture. For additional information, refer the *GTP-C Path Failure Enhancements and Improved Debugging Tools* section in the *P-GW Administration Guide*.

GTPP Suppress-CDR No Zero Volume

This feature allows suppression of CDRs with zero byte data count, so that the OCG node is not overloaded with a flood of CDRs. The CDRs can be categorized as follows:

- Final-cdrs: These CDRs are generated at the end of a context.
- Internal-trigger-cdrs: These CDRs are generated due to internal triggers such as volume limit, time limit, tariff change, or user-generated interims through the CLI commands.
- External-trigger-cdrs: These CDRs are generated due to external triggers such as QoS Change, RAT change and so on. All triggers which are not considered as final-cdrs or internal-trigger-cdrs are considered as external-trigger-cdrs.

The customers can select the CDRs they want to suppress.

The CLI command mentioned below helps suppress CDRs on different CDR triggers supported in CUPS:

- **[default | no] gtpp suppress-cdrs zero-volume { external-trigger-cdr | final-cdr | internal-trigger-cdr }**

Location Based DNS and PCSCF IP Address Selection

Location-based DNS and P-CSCF Selection provides an option to the operator to manage the DNS server address and P-CSCF IP address according to location information.

P-GW gathers the DNS server address and P-CSCF IP address information by Tracking Area Identifier (TAI), which is achieved through the TAC-based Virtual APN (VAPN) selection.

When UE sends the PCO request in session creation, P-GW selects the Virtual APN (VAPN) with the received location information. The selected VAPN (with DNS server address and P-CSCF IP address configured in it) with PCO IE is sent in the Create session response.

Following are the CLI commands for enabling the Location-based DNS and PCSCF IP address selection:

Command	Description
Tracking-area-code-range from <*start value*> **to** <*end value*>	Provides the tracking area code range, starting from 0 through 65536. The end value is always greater than the start value.
P-cscf priority <*priority*> **ip/ipv6** <*IPv4/Ipv6 address*>	Specifies the priority for P-CSCF address for the APN. Address_ priority is an integer 1–3. One is the maximum priority. IPv4_address is in IPv4 dotted-decimal notation. IPv6_address is in IPv6 colon-separated-hexadecimal notation.
Show apn name <*APN Name*>	To show PCSCF IP address at APN

Command	Description
dns primary <*IPv4 address*> **Dns secondary** <*IPv4 address*> **ipv6 dns primary** <*IPv6 address*> **ipv6 dns secondary** <*IPv6 address*>	Primary: Configures the primary DNS server for the APN. Secondary: Configures the secondary DNS server for the APN. Only one secondary DNS server is configurable. Address: Configures the IP address of the DNS server expressed in IPv4 dotted-decimal notation. Default: primary = 0.0.0.0, secondary = 0.0.0.0 dns_address: Specifies the IP address of the DNS server to remove, expressed in IPv4 dotted-decimal notation.
Show apn name <*APN Name*>	To show DNS IP address at APN

MPRA Support

P-GW supports negotiation of Multiple-Presence Reporting Area feature in Feature-List-ID 2 over Gx interface with PCRF. The CNO-ULI feature works only when the P-GW and/or the PCRF doesn't support Multiple-PRA and both P-GW and PCRF support CNO-ULI.

For Multiple-PRA feature support during the lifetime of the IP-CAN session, P-GW handles the change of UE Presence in Reporting Areas request from PCRF in PRA-Install AVP including the Presence-Reporting-Area-Information AVPs. Each AVP contains the Presence Reporting Area Identifier within the Presence-Reporting-Area-Identifier AVP

For more information on Presence Reporting Area (PRA) and Multiple PRA, refer the *Presence Reporting Area* chapter in the *StarOS P-GW Administration Guide*.

No udp-checksum Support

This feature supports **no udp-checksum** CLI command for CUPS under GTPU service where **udp-checksum** is disabled in the outer GTPU header for the downlink subscriber packet. When downlink packet arrives from internet, the GTPU header is added on top of the packet and is sent to the access side. The "checksum" value is zero in the outer UDP layer of this packet enabling optimization and therefore, improving the performance throughput.

Use the following configuration to enable the feature.

```
configure
  context context_name
    gtpu-service gtpu_service_name
    [ no ] udp-checksum
    end
```

Show Commands and Outputs

This section provides information about the show CLI commands available in support of the feature.

Use the following command to determine if **GTPU UDP Checksum** is enabled or disabled.

- **show gtpu-service all**—Displays all GTPU services.
- **show gtpu-service name** *service_name*—Displays information for the specific GTPU service name.

QUIC IETF Implementation

In the current framework, Deep Packet Inspection (DPI) is done for every packet in a flow when it reaches the plugin. The DPI is done by analyzing the packets and extracting deterministic patterns. The DPI is done in-order to detect the application and to classify its subtype. Plugin excludes the flow after the DPI. The flow is offloaded after the detection. As part of QUIC IETF, the initial QUIC handshake packets (Client/Server Hello) are encrypted over the network. Hence, there are no deterministic patterns available for detection of the application. Support is added in p2p plugin to decrypt and obtain the SNI (Server Name Indication) for detection.

Configuring QUIC IETF

Use the following configuration to enable or disable the QUIC IETF decryption.

```
configure
  active-charging service acs_service_name
    p2p-detection debug-param protocol-param p2p_quic_ietf_decrypt 1
    end
```

Note By default, the CLI is disabled and there's minimal impact on the performance due to TLS decryption.

Optimization for egtpinmgr Recovery

Previously, when the egtpinmgr task restarted, it took a significant amount of time for it to recover. As a result, the outage time when the SAEGW were unable to accept any new calls during egtpinmgr recovery was high.

The software has been enhanced to optimize the recovery outage window in the event of an egtpinmgr task restart; this has been achieved by optimizing the internal algorithms of egtpinmgr recovery and the data structures required. In addition, recovery time now is dependent only on the number of unique IMSIs and not on the number of sessions for an IMSI.

Note The Optimization for egtpinmgr Recovery is an existing feature that is supported in non-CUPS architecture. With this release, the feature is qualified in CUPS architecture.

Quota Hold Time Support

Quota-Hold-Time (QHT) is an inactivity time duration, after which the Gateway(Diameter client) returns the Charging-Bucket with its usage and reaches a clean-state.

The QHT value is provided by the OCS per Category - Multiple-Services-Credit-Control (MSCC), or the gateway provides an option to configure the default value of the QHT - for enabling the default QHT value for the MSCCs for which the OCS has not provided any QHT AVP.

The QHT timer runs per MSCC bucket. If the QHT timer expires without a packet during run-time, then the usage is reported with the Reporting-Reason: QHT as per 3GPP specification.

The QHT value received in the CP from the OCS, is sent in the "Quota Holding Time" IE defined in the CUPS specification 3GPP TS 29.244. Also along with provisioning the Quota-Holding-Time IE to the UP, the

Reporting-Triggers will be sent with the bit corresponding to Quota-Holding-Time SET, so that on QHT expiry the reporting takes place.

The UP on receiving the Quota-Holding-Time IE along with the QHT Reporting-Triggers enabled, starts the timer per URR to monitor the inactivity period. Once the inactivity period exceeds the QHT time, the Usage-Reporting is initiated from the UP for the Trigger : Quota-Holding-Time.

The CP on receiving the QHT event from UP, triggers the QHT reporting to the OCS after updating the usage in the MSCC bucket.

Configuring Quota Hold Time

Use the following configuration to enable Quota Hold Time in CUPS:

```
configure
  require active-charging
  active-charging service service_name
    credit-control group group_name
      quota-hold-time timer_value
      end
```

NOTES:

- **quota-hold-time**: configures the inactivity duration after which the charging bucket reports its usage and have a clean state.

Limitation

The QHT (inactivity-timer) usually is a larger value compared to the flow-idle timer. If the flow-idle timer is larger than QHT, then there is a possibility for the flows present even after the QHT expiry, and is processed by VPP as per the NoQuota Pending-Traffic-Treatment configuration.

S-GW Paging Enhancement

S-GW Paging includes the following scenarios:

Scenario 1: S-GW sends a Downlink Data Notification (DDN) message to the MME/S4-SGSN nodes. MME/S4-SGSN responds to the S-GW with a DDN Ack message. While waiting for the DDN Ack message from the MME/S4-SGSN, if the S-GW receives a high priority downlink data, it does not resend a DDN to the MME/S4-SGSN.

Scenario 2: If a DDN is sent to an MME/S4-SGSN and TAU/RAU MBR is received from another MME/S4-SGSN, S-GW doesn't send DDN.

Scenario 3: DDN is sent to an MME/S4-SGSN and DDN Ack with Cause #110 is received. DDN Ack with cause 110 is treated as DDN failure and standard DDN failure action procedure is initiated.

To handle these scenarios, the following two enhancements are added to the DDN functionality in CUPS architecture:

- High Priority DDN at S-GW
- MBR-DDN Collision Handling

These enhancements support the following:

- Higher priority DDN on S-GW and SAEGW, which helps MME/S4-SGSN to prioritize paging.

- Enhanced paging KPI and VoLTE services.
- DDN message and mobility procedure so that DDN isn't lost.
- MBR guard timer, which is started when DDN Ack with temporary HO is received. A CLI command **ddn temp-ho-rejection mbr-guard-timer** has been introduced to enable the guard timer to wait for MBR once the DDN Ack with cause #110 (Temporary Handover In Progress) is received.
- TAU/RAU with control node change triggered DDNs.

In addition, to be compliant with 3GPP standards, support has been enhanced for Downlink Data Notification message and Mobility procedures. As a result, DDN message and downlink data which triggers DDN is not lost. This helps improve paging KPI and VoLTE success rates in scenarios where DDN is initiated because of SIP invite data.

Note For information on Downlink Data Notification (DDN) messages with support for DDN Delay and DDN Throttling, refer the *SAEGW Idle Buffering with DDN Delay and DDN Throttling* chapter in this guide.

For more information on how S-GW Paging Enhancement feature works, configuration, monitoring and troubleshooting, refer the *S-GW Paging Enhancements* chapter in the StarOS *S-GW Administration Guide*.

Session Recovery in User Plane

Support is added to recover the Session Manager process in the event of any crash. The recovered Session Manager has all the existing subscriber session on the recently crashed Session Manager process.

Uplink and Downlink data flow is processed on the newly recovered Session Manager process for all recovered subscriber sessions.

SRVCC PS to CS Handover Indication and the QoS Class Index IMS Media Configuration Support

This feature notifies the PCRF about the cause for PCC rule deactivation on Voice bearer deletion. This notification helps the PCRF to take further action appropriately.

This feature ensures the compliance for SRVCC. This feature also supports the PS-to-CS handover indication after release of the voice bearers.

SRVCC service for LTE lets a single radio User Equipment (UE) accessing IMS-anchored voice call services to switch from LTE network to Circuit Switched domain. The UE switches the network while it can transmit or receive on only one of the access networks then. The SRVCC service removes the need for a UE to have multiple Radio Access Technology (RAT) capabilities.

After handing over the PS sessions to the target, the source MME removes the Voice Bearers (VB). The MME removes the VB by deactivating the voice bearers. The MME bars the VB towards S-GW/P-GW and sets the VB flag of Bearer Flags IE in the Delete Bearer Command message (TS 29.274 v9.5.0).

If the IP-CAN bearer termination happens due to PS to CS handover. The PCEF reports the related PCC rules for this IP-CAN bearer by including the Rule-Failure-Code AVP set to the value: PS_TO_CS_HANDOVER (TS 29.212 v10.2.0 and TS 23.203 v10.3.0).

Support for new AVP PS-to-CS-Session-Continuity (added in 3GPP Release 11) inside Charging Rule Install indicates the bearer support for PS to CS continuity.

QCI IMS-Media Configuration Support

Specifies the QoS Class Index (QCI) value to mark the IMS media bearers for preferential treatment during session recovery and ICSR switchover.

Mode

Exec > Global Configuration > Context Configuration > APN Configuration

configure > context <*context_name* > **apn** <*apn_name*>

Entering the above command sequence results in the following prompt:

[context_name]host_name(config-apn)#

Syntax

qci *value_bytes* **ims-media**

no qci *value_bytes* **ims-media**

Note

- *no*: Disables this IMS QCI feature.
- *ims_media*: Marks bearers classified as IMS media for preferential treatment during session recovery and ICSR switchover.
- *value_bytes*: Specifies the QCI value an integer from 1 through 254.

Usage Guidelines

Use this command to specify the QCI value to be used to mark bearers classified as IMS media for preferential treatment during session recovery and ICSR switchover.

The following prerequisites apply to the implementation of this feature:

- A dedicated APN must be reserved for VoLTE traffic.
- A call connected to this APN will not be classified as Active VoLTE unless there is a dedicated bearer matching the VoLTE-configured QCI.
- Preferential treatment would be given to only those calls which are active VoLTE.
- A GGSN call connected to this APN will not be classified as Active VoLTE unless there is network initiated bearer matching the VoLTE-configured QCI.
- VoLTE marking is preserved across a Gn-Gp handoff.

When this feature is enabled via a CLI command, the actions are taken:

- During bearer creation
  - New bearer QCI is matched against APN configuration.
  - If the QCI matches an APN configuration, the bearer is marked for preferential treatment.
  - Flow_entries are modified with this information (if this is first VoLTE bearer).

- Egtpu_session is updated with the VoLTE tag during a rx_setup request.
- An indication message informs ECS about the VoLTE tagging.

- During bearer deletion
  - Flow_entry is updated with VoLTE information if this is the last VoLTE bearer.
  - ECS is informed of the deletion via an indication message.

The following command enables preferential treatment for IMS bearers with a QCI of 9:

qci 9 ims-media

Support for ip hide-service-address CLI Command

The **ip hide-service-address** CLI command is supported in CUPS.

When enabled, this CLI renders the IP address of the GGSN unreachable from mobile stations (MSs) using this APN. This command is configured on a per-APN basis.

Use the following configuration to enable or disable the feature.

```
configure
  context context_name
    apn apn_name
      [ default | no ] ip hide-service-address
      end
```

- **default**: Does not allow the mobile station to reach the GGSN IP address using this APN.
- **no**: Allows the mobile station to reach the GGSN IP address using this APN.
- Use this command to prevent subscribers from using traceroute to discover the network addresses that are in the public domain and configured on services.

Support for regardless-of-other-triggers CLI Command

This feature supports **regardless-of-other-triggers** option in CLI for CUPS. **regardless-of-other-triggers** option enables eG-CDR or P-GW-CDR generation at the fixed time interval irrespective of any other eG-CDR or P-GW-CDR triggers that may occur in between. Therefore, when you enable this option although other CDR triggers occur, the Time Limit CDR gets triggered dynamically at every *interval* in seconds, that is, the Time Threshold calculation is based on the sum of the last threshold time and the interval. This option supports session recovery and ICSR.

Use the following configuration to enable the feature.

```
configure
  active-charging service service_name
    rulebase rulebase_name
      egcdr threshold interval interval regardless-of-other-triggers
      end
```

The following steps are carried out when a new call comes in:

- When you enable, **regardless-of-other-triggers** even if any other usage report triggers in between, timer will not be reset, and the session usage report for the time threshold occurs for every interval time configured.

Show Commands and Outputs

This section provides information about the show CLI commands available in support of the feature.

- **show active-charging rulebase name** *name*
- **show active-charging rulebase all**

The output of these CLI commands includes the following fields to support this feature.

- Interval Threshold: <seconds> (secs) Regardless of Other Triggers

TFT Suppression for Default Bearer

Feature Description

TFT Suppression for default bearer is supported in the UPC CUPS architecture. Following CLI commands are added in support of this feature.

- **policy-control update-default-bearer**
- **no tft-notify-ue-def-bearer**

The preceding CLI commands are used to bind all the predefined rules received from PCRF without QoS and ARP or with the same QoS and ARP as that of the default bearer, to the default bearer.

Important This CLI is applicable to all the rulebases in the chassis configuration. If the rulebase is changed to some other rulebase in the interim period or anytime later, this CLI will continue to apply to the current new rulebase too.

Configuring TFT Suppression

Configuring TFT Suppression in Default Bearer for Predefined Rules

Use the following commands to configure TFT Suppression for default bearers.

```
configure
  require active-charging
  require active-charging service_name
    [ default | no ] policy-control update-default-bearer
    end
```

Caution Upon executing this CLI command "**no policy-control update-default-bearer**", system crash is likely to occur if the TFT information is not added to the charging-action.

Configuring TFT Suppression in Default Bearer

Use the following commands to configure TFT Suppression for default bearers.

```
configure
   require active-charging
   require active-charging service_name
     rulebase rulebase_name
        [ default | no ] tft-notify-ue-def-bearer
     end
```

Note

- **default:** Configures this command with its default setting.

  Disables only binding those rules having QoS of default bearer to the default bearer and specifies to not ignore other rules. Rules having respective QoS gets attached to the relevant bearers. Also, TFT updates towards UE (access side) is not suppressed.

- **no:** Enables binding rules having QoS of default bearer to the default bearer and specifies to ignore other rules.

  In case no QoS is specified the rule gets attached to default bearer. Also, TFT updates towards UE (access side) is suppressed for default bearer. So only one default-bearer is ever be created.

Zero-byte EDR Suppression

The Zero-byte Event Data Record (EDR) Suppression, a CLI-controlled feature, enables or disables creation of EDRs when there is no data for the flow. A zero-byte EDR is typically possible when two successive EDRs are generated for a flow. The CLI command suppresses the second such EDR for the flow.

Use the following configuration to enable or disable the suppression of zero-byte EDRs.

```
configure
   active-charging service service_name
     rulebase rulebase_name
        [ default | no ] edr suppress-zero-byte-records
        end
```

NOTES:

- **default**: Configures this command with its default setting.

  Default: Disabled; same as **no edr suppress-zero-byte-records**

- **no**: Disables the suppression of zero-byte EDRs.
- **edr suppress-zero-byte-records**: Suppresses zero-byte EDRs.
- The “Total zero-byte EDRs suppressed” field in the output of the following CLI command can be used to verify if the zero-byte EDRs are suppressed: **show user-plane-service statistics rulebase name** *rulebase_name*.

How It Works

This section describes the Call Flows for User Plane service.

Call Flows

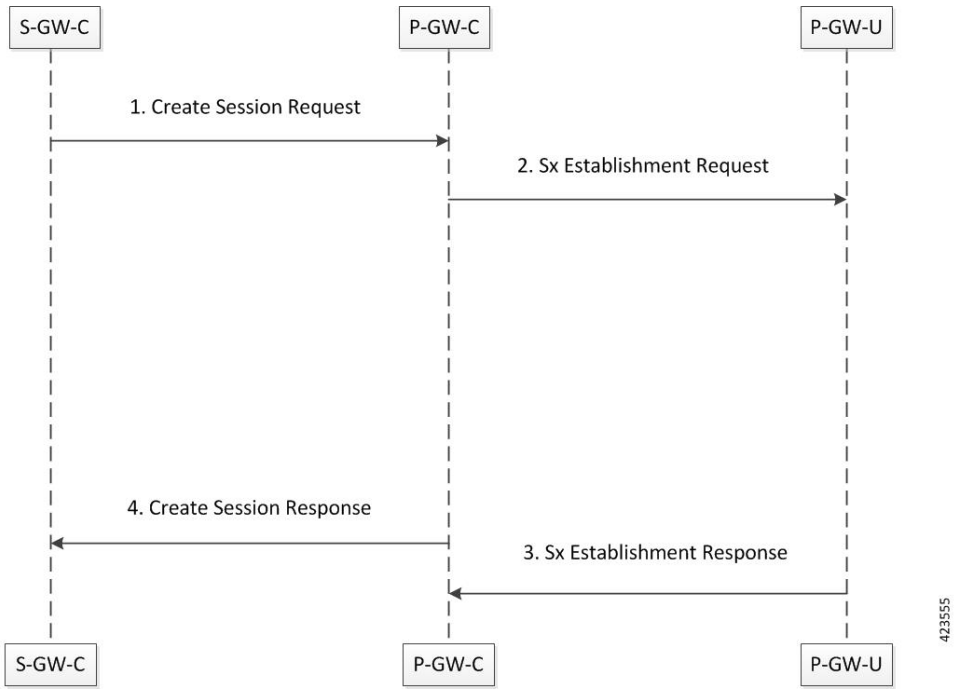
This section describes the User Plane Call Flows in the CUPS architecture.

P-GW Data Session

This section describes the P-GW initial attach procedure.

Initial Attach Procedure (Pure P)

Following call flow illustrates, at a high-level, the initial attach procedure for a Pure-P PDN.

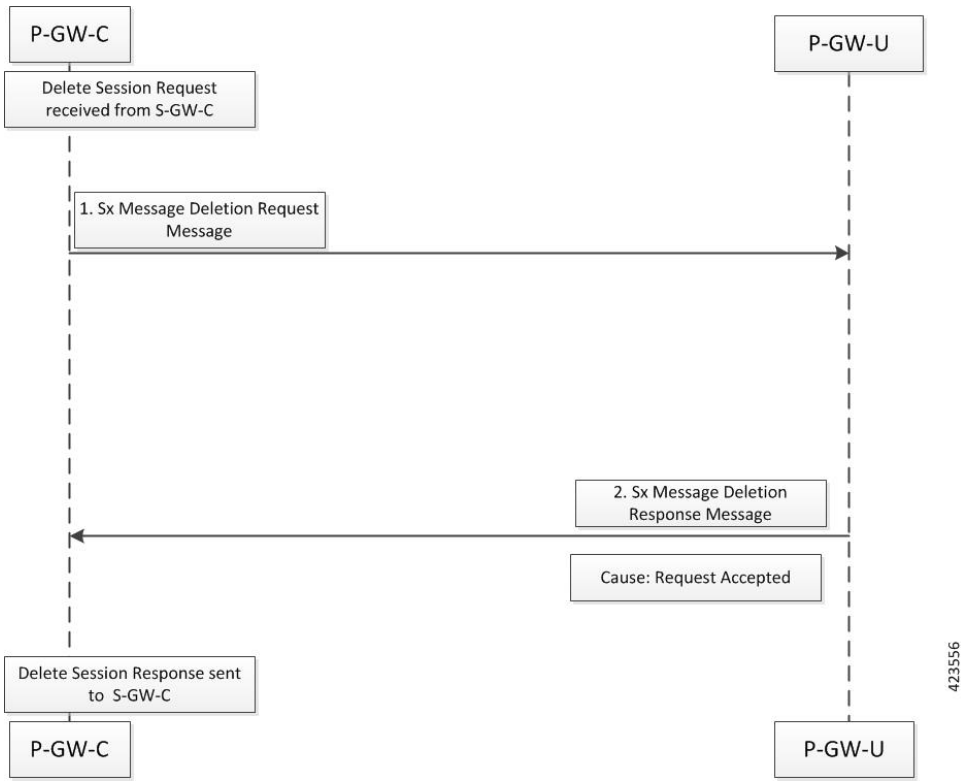


- P-GW receives a Create Session Request message including an APN, on the S5/S8 interface.
- P-GW-C initiates an Sx establishment request on Sxb interface towards selected P-GW-U with PRDs, FARs information to establish the data path. PGW-C does not support TEID (Tunnel Identifier) allocation; it is allocated by PGW-U.
- Once the resources are allocated (TEID and so on), P-GW-U sends an Sx establish response message towards P-GW-C.
- P-GW responds to the S-GW with a Create Session Response message including the assigned address, TEID, and additional information.

- The S5/S8 data plane tunnel is established and the PGW-U can forward and receive packets to and from the PDN.

Initial Detach Procedure (Pure P)

Following call flow illustrates, at a high-level, the initial detach procedure for a Pure-P PDN.



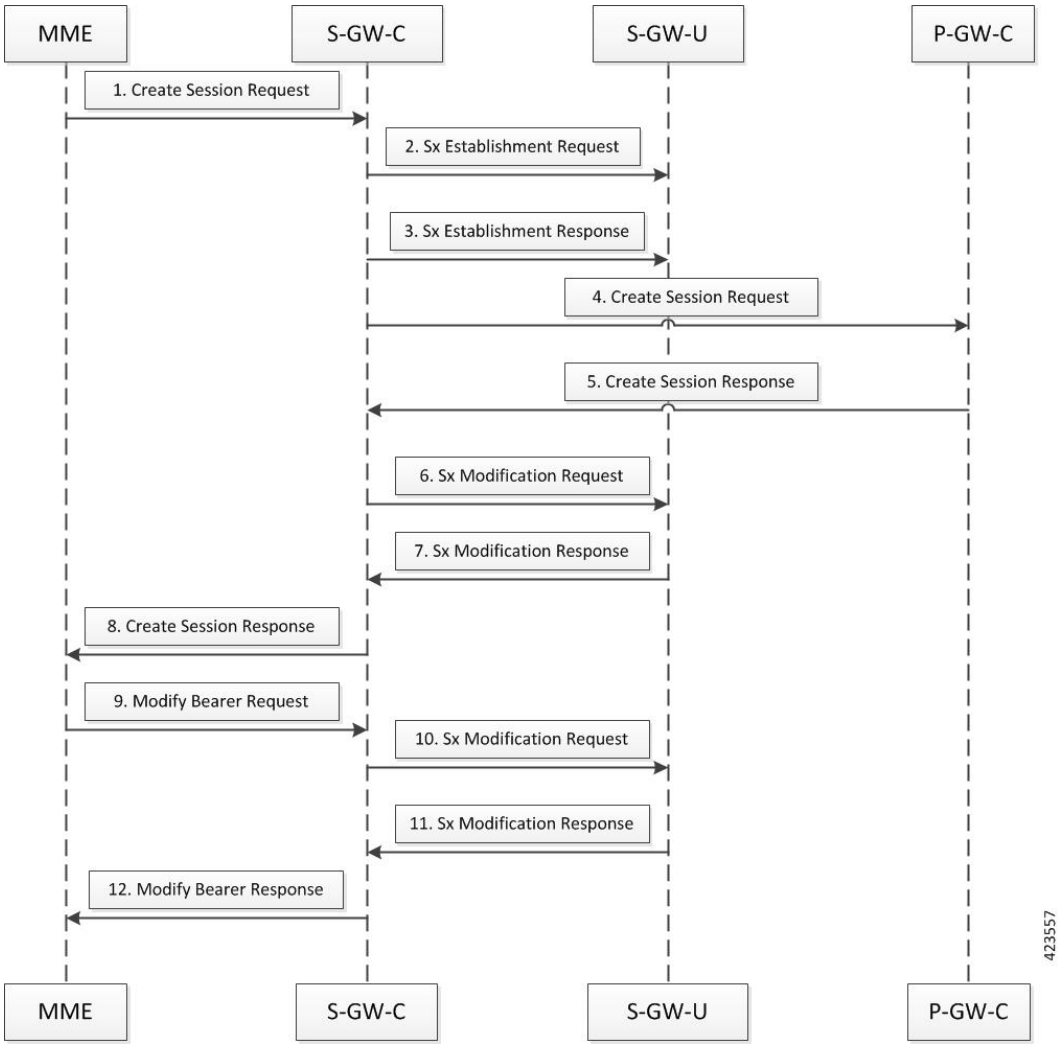
- P-GW receives a Create Session Request message including an APN, on the S5/S8 interface.
- P-GW-C initiates an Sx establishment request on Sxb interface towards selected P-GW-U with PRDs, FARs information to establish the data path. PGW-C does not support TEID (Tunnel Identifier) allocation; it is allocated by PGW-U.
- Once the resources are allocated (TEID and so on), P-GW-U sends an Sx establish response message towards P-GW-C.
- P-GW responds to the S-GW with a Create Session Response message including the assigned address, TEID, and additional information.
- The S5/S8 data plane tunnel is established and the PGW-U can forward and receive packets to and from the PDN.

S-GW Data Session

This section describes the S-GW initial attach procedure.

Initial Attach Procedure (Pure S)

Following call flow illustrates, at a high-level, the initial attach procedure for a Pure-S PDN.

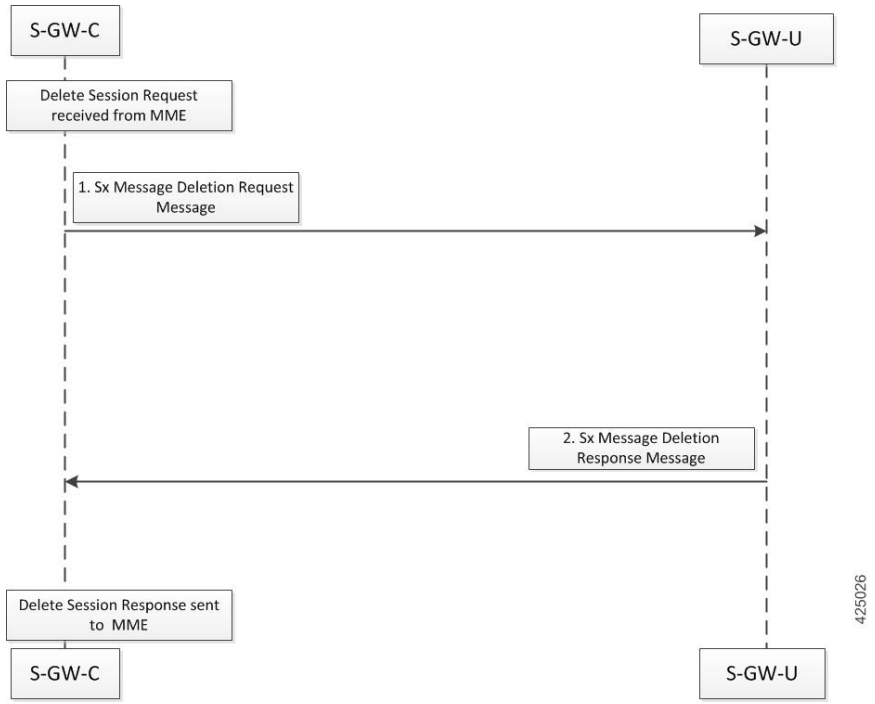


- On S11 interface, S-GW-C receives a Create Session Request message including an Access Point Name (APN) from MME.
- S-GW-C initiated the Sx establishment request on the Sxa interface towards the selected S-GW-U with PDRs, FARs information to establish data path. Here, the S-GW-C does not support the TEID (Tunnel Identifier) allocation. It is allocated by the S-GW-U.
- After allocation of resources such as egree TEID and so on, S-GW-U sends the Sx establishment response message towards the S-GW-C.
- The S-GW-C initiates the Create Session Request towards the selected P-GW-C.
- P-GW-C responds with the Create Session Response with the IP address and default bearer related information.
- SGW-C initiates Sx Modification Request message towards SGW-U to update FAR (Forwarding action) information for the existing session.

- SGW-U provides Sx Modification Response with success after updating information.
- SGW-C sends Create Session Response with all necessary information for Default Bearer towards MME.
- MME initiates Modify Bearer Request message towards SGW-C, once it received eNodeB's F-TEID information.
- SGW-C initiates Sx Modification Request towards SGW-U for updating FAR information on eNodeB's F-TEID.
- After successfully updating, Sx Modification Response is sent to SGW-C.
- SGW-C inturn will send Modification Response message towards MME to complete attach procedure.
- SGW-U has established S1U side data tunnel towards eNodeB and S5/S8 side data tunnel towards PGW-U. Now, SGW-U can forward and receive packets to/from PGW as well eNodeB.
- The S5/S8 data plane tunnel is established and the PGW-U can forward and receive packets to/from the PDN.

Initial Detach Procedure (Pure S)

Following call flow illustrates, at a high-level, the initial detach procedure for a Pure-S PDN.



- Once Delete Session Request is received from MME, SGW initiate Sx Delete Request message towards SGW-U.
- SGW-U clears all allocated User-Plane resources and responds back with Cause Success to SGW-C.
- SGW-C responds back to MME with Delete Session Response message.

Support for Addition, Deletion and Updation of Dedicated Bearers for S-GW

Feature Description

Addition, Deletion and Updation of Dedicated Bearers for Pure-S calls is supported in the CUPS architecture.

The following functionality is added in support of this feature:

- SAEGW-CP supports Create Bearer Request for dedicated bearer for Pure-S Call.
- SAEGW -CP supports multiple bearer contexts in single Create Bearer Request.
- SAEGW-CP supports multiple Create Bearer request in parallel for different PDN; these PDN can be Pure-S PDN or Collapsed and Pure-S combinations.
- SAEGW-UP creates uplink and downlink bearer stream at VPP for Pure-S call per bearer. Number of streams per direction depends on the GTP-U Service IP address.
- SAEGW-CP supports Release Access Bearer Request (RAB) with dedicated bearer, all FAR corresponding to all bearer is modified.
- SAEGW-CP supports Modify Bearer Request (Idle mode, Connected mode) with dedicated bearer.
- SAEGW-CP supports Create Bearer Response Failure handling from MME.
- SAEGW-CP and SAEGW-UP supports DSCP marking for default and dedicated bearer with VPP.
- SAEGW-CP and SAEGW-UP supports Delete Bearer Request for dedicated bearer. SAEGW-UP removes bearer stream and TEP entries belonging to those bearers.
- SAEGW-CP supports Pure-S Dedicated Bearer Creation when call is in IDLE state.
- SAEGW-CP supports Pure-S Dedicated Bearer S-GW Relocation (both X2 and S1-based).
- SAEGW-CP supports Pure-S Dedicated Bearer Update success scenarios.
- SAEGW-CP supports Piggybacking of Create Bearer Request for dedicated bearer for Pure-S call along with Create Session Response.
- SAEGW-CP supports Piggybacking of Create Bearer Response for dedicated bearer for Pure-S call with Modify Bearer Request.
- SAEGW-CP supports Pure-S Dedicated Bearer Creation if P-GW receives bearer creation as part of CCA-I, where P-GW does not send Piggyback request, which results in Create Session Response followed by Create Bearer Request.
- SAEGW-CP supports Session Recovery and ICSR with Pure-S dedicated bearer.
- SAEGW-CP supports Create Bearer Request and Delete Bearer Request (default bearer) collision.
- SAEGW-CP supports Create Bearer Request and Delete Session Request collision.
- SAEGW-CP supports Create Bearer Response and Delete Bearer Request (default bearer) collision.
- SAEGW-CP supports Create Bearer Response and Delete Session Request collision.
- SAEGW-CP supports End Marker with Pure-S default and dedicated bearer.
- SAEGW-UP supports Session Recovery with Pure-S default and dedicated bearer.

- SAEGW-UP supports movement of IP transport from IPv4 to IPv6, or IPv6 to IPv4, during IDLE->Active and Handover procedure on S1U interface. Transport selected on S1U at the time of Attach is supported. For example, eNode handover from IPv4 eNodeB to IPv6 eNodeB will work.
- SAEGW-CP supports CBRsp with Cause Partially Accepted and Context Not Found.
- SAEGW-CP supports Downlink Data Notification for Pure-S Call, so when UE moves to IDLE state for Pure-S call, FAR action is set as BUFFER.
- SAEGW-CP supports Update Bearer Response with cause PARTIALLY_ACCEPTED and context not Found.
- SAEGW-CP supports the Error and Failure handling from other peer nodes including User Plane node.

Limitations

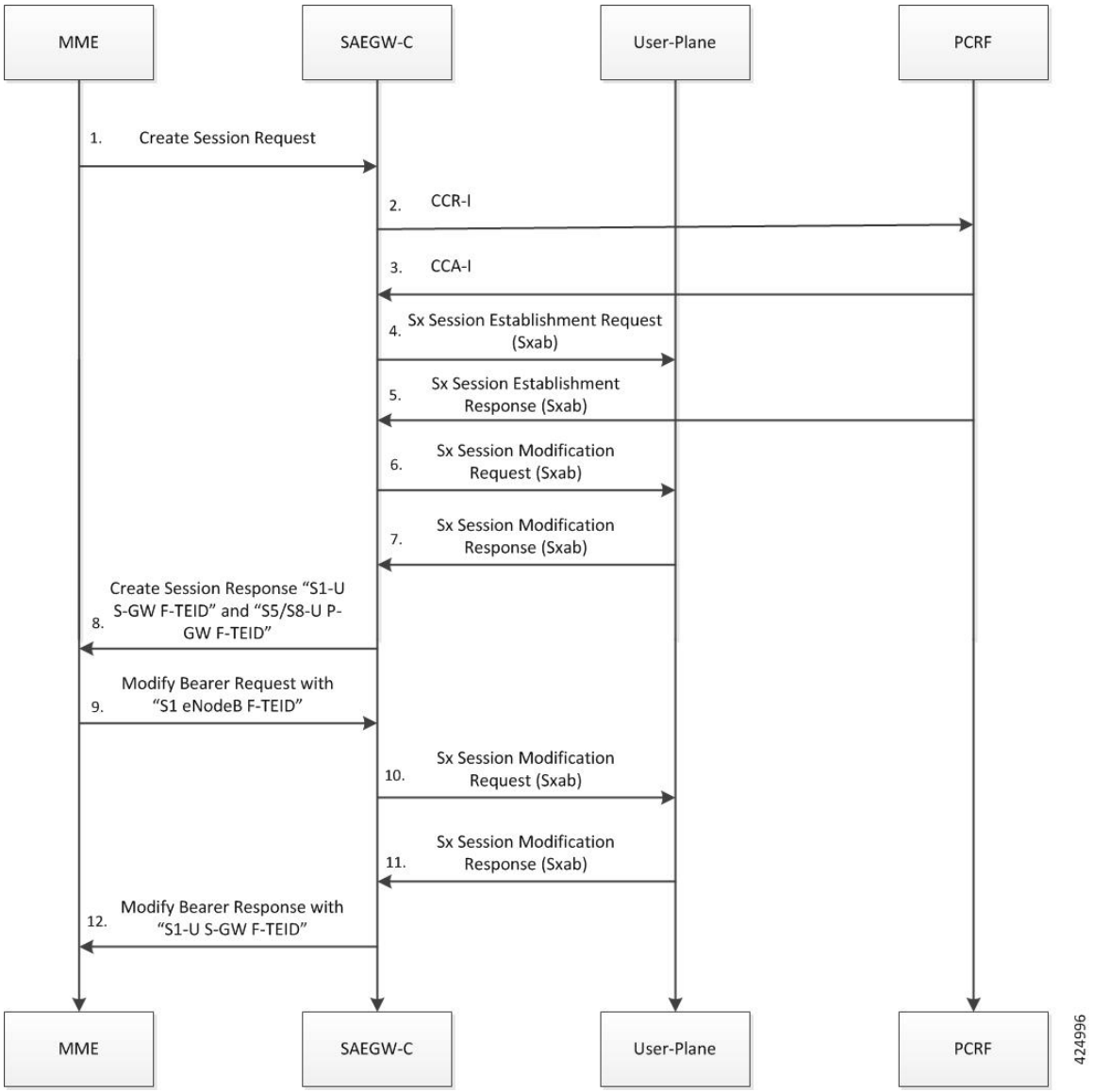
For Pure-S calls, Idle Session timeout is not supported.

Support for Collapse Call

Following call flow illustrates, at a high-level, the detach procedure for UE initiated Collapsed PDN.

Initial Attach Procedure (Collapsed PDN)

The following call flow illustrates, at a high-level, the initial attach procedure for Collapsed PDN.



1. For CUPS SAEGW collapsed call, SAEGW-C does the following:
   - After Gx interaction, performs Gx communication (CCR-I and CCA-I).
   - Performs User Plane selection based on **user-plane-profile** configured with IP Pool (APN associated with IP Pool).
   - Establishes GTP-U session (required for RA/RS, in case of IPv6/IPv4v6 PDN).
   - Performs Sxab interaction with selected User Plane.
2. Sx Establishment Request contains the following information:
   - Create PDR/FAR information for S-GW Uplink and Downlink data path (Sxa Type PDR).
   - Create PDR/FAR/URR information for Uplink and Downlink data path (Sxb Type PDR): For dynamic/pre-defined/static rules.

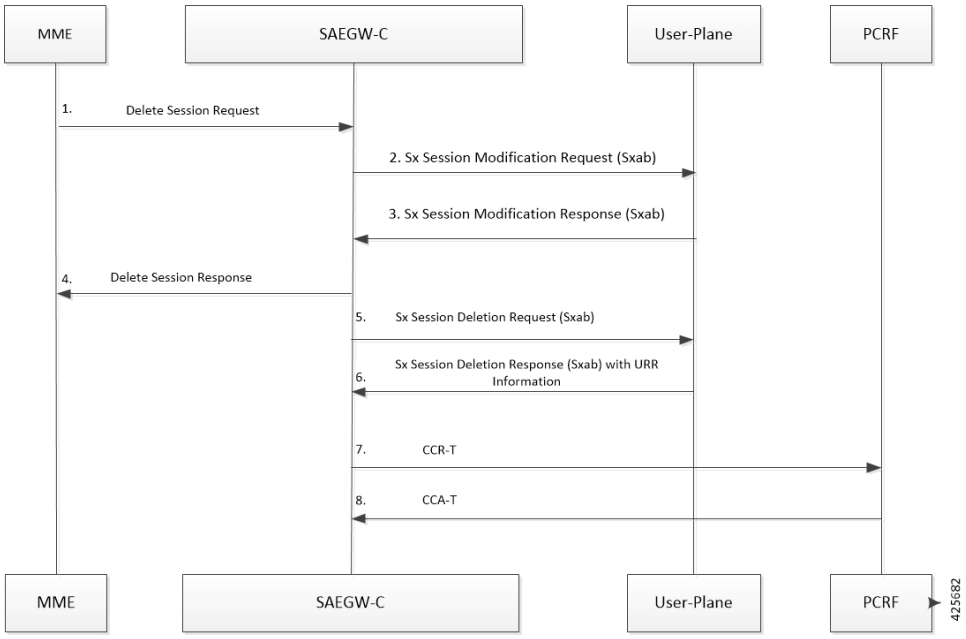
- Create PDR/FAR for RA/RS (Sxb Type PDR): Required for IPv6/IPv4v6 PDN Type.
- Additionally, Control Plane requests User Plane to allocate F-TEID for:
  - S-GW Ingress "S1-U S-GW F-TEID",
  - S-GW Egress "S5/S8-U S-GW F-TEID", and
  - P-GW Ingress PDR "S5/S8-U P-GW F-TEID"

3. User Plane provides following information as part of Sx Session Establishment Response:
   - Created PDR: S-GW Ingress PDR "S1-U S-GW F-TEID",
   - Created PDR: S-GW Egress PDR "S5/S8-U S-GW F-TEID", and
   - Created PDR: P-GW Ingress PDR "S5/S8-U P-GW F-TEID"
4. On receipt of successful Sx Session Establishment Response, the Control Plane triggers Sx Modification Request with the following information:
   - To update P-GW (Sxb) "Uplink PDR" with "Outer Header Removal" based on IP address information in "S5/S8-U S-GW F-TEID"
   - To update P-GW (Sxb) "Downlink FAR" with "Outer Header Creation" as "S5/S8-U S-GW F-TEID"
   - To update S-GW (Sxa) "Uplink FAR" with "Outer Header Creation" as "S5/S8-U P-GW F-TEID"
   - To update S-GW (Sxa) "Downlink PDR" with "Outer Header Removal" based on IP address information in "S5/S8-U P-GW F-TEID"
5. On receipt of Sx Session Modification Response, the SAEGW-C sends Create Session Response toward MME with "S1-U S-GW F-TEID" and "S5/S8-U P-GW F-TEID".
6. On receipt of Modify Bearer Request (MBR), the SAEGW-C does the following:
   - Trigger Sx Session Modification Request:
     - To update Downlink FAR with "Outer Header Creation" as "S1 eNodeB F-TEID".
     - To update Uplink PDR with "Outer Header Removal" based on IP address information in "S1 eNodeB F-TEID".
7. On receipt of Sx Session Modification Response, the SAEGW-SGW-C sends MBR with "S1-U S-GW F-TEID".

Initial Detach Procedure (Collapsed Call)

Detach Procedure (Collapsed): UE Initiated

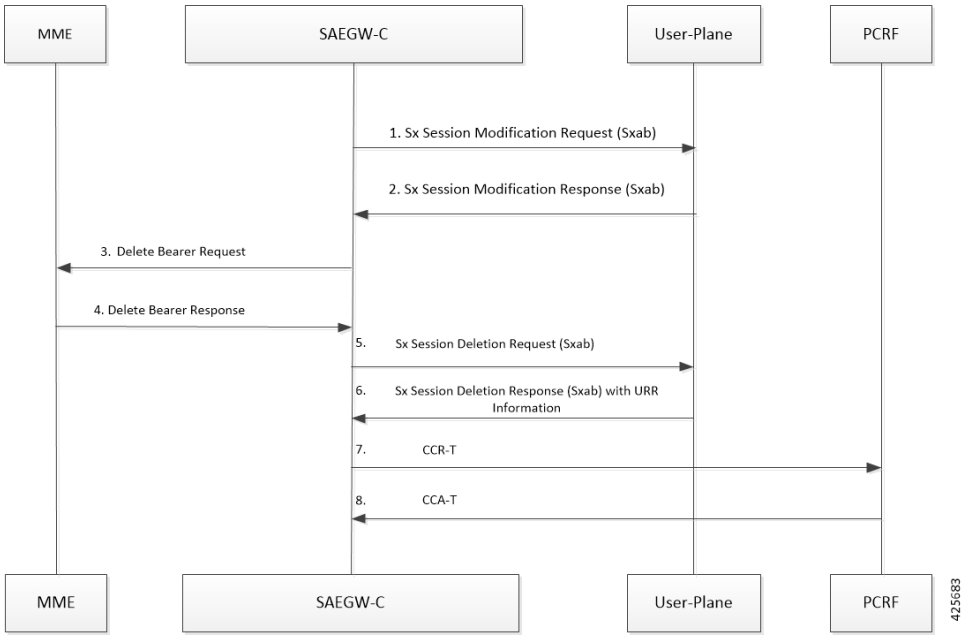
Following call flow illustrates, at a high-level, the detach procedure for UE initiated Collapsed PDN.



1. On receipt of Delete Session Request, the SAEGW-C performs Sxab interaction to update FAR with Apply Action as "DROP" for both Uplink and Downlink data path.
2. On receipt of Sx Session modification Response, SAEGW-C sends Delete Session Response towards MME.
3. For CUPS SAEGW Collapsed call, the SAEGW-C does the following:
   - Removes GTP-U session (required for RA/RS in case of IPv6/IPv4v6 PDN).
   - Performs Sxab interaction with the selected User Plane.
4. On receipt of Sx Session Deletion Response, the SAEGW-C does the following:
   - Performs Gx communication (CCR-T and CCA-T).
   - Generates CDR (Gz) based on URR information received.

Detach Procedure (Collapsed): Network Initiated

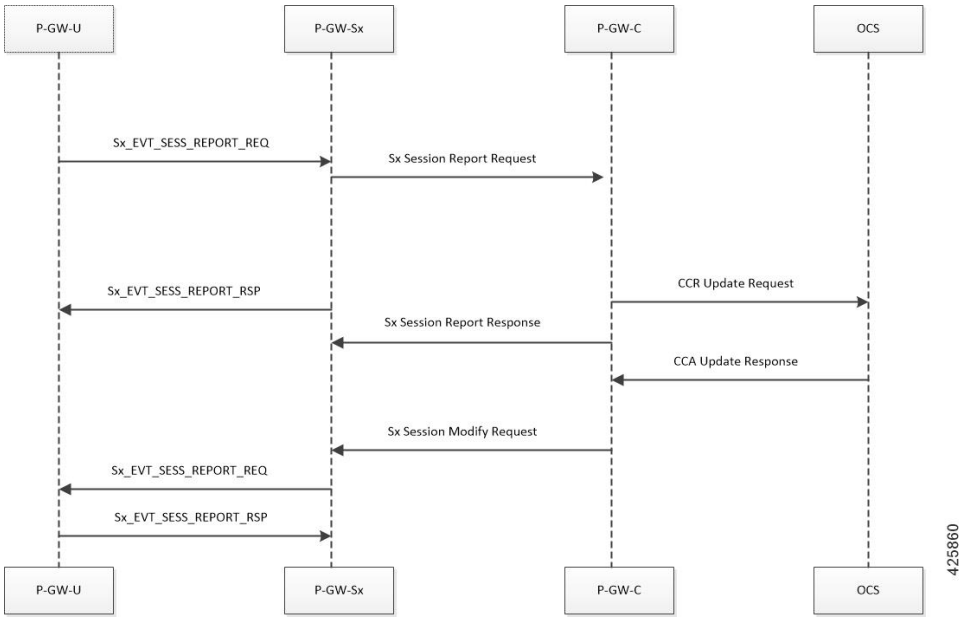
Following call flow illustrates, at a high-level, the detach procedure for network initiated Collapsed PDN.



1. On receipt of Delete Bearer Request (RAR initiated or by the **clear sub all** CLI), SAEGW-C performs Sxab interaction to update FAR with Apply Action as "DROP" for both Uplink and Downlink data path.
2. On receipt of Sx Session Modification Response, SAEGW-C sends Delete Bearer Request toward MME.
3. For CUPS SAEGW Collapsed call, the SAEGW-C does the following:
   - Removes GTP-U session (required for RA/RS in case of IPv6/IPv4v6 PDN).
   - Performs Sxab interaction with the selected User Plane.
4. On receipt of Sx Session Deletion Response, the SAEGW-C does the following:
   - Performs Gx communication (CCR-T and CCA-T).
   - Generates CDR (Gz) based on URR information received.

P-GW Session Reporting with Gy Interface

This section describes P-GW session reporting with Gy interface.



URR Support in Session Establishment Request

- User plane module supports the storage of a list of URRs received as part of session establishment request.
- Each PDR is associated with one or more URRs.
- A particular URR is linked to another URR.
- Each URR contains the measurement method (time or volume), and reporting triggers that indicates the event on which the user plane has to send usage report.
- The URR have both volume-quota and volume-threshold present for the Gy-URRs.

Session Delete Response

This message sent from the User Plane is in response to a session deletion request from control plane. This results in the termination of the Sx session at User plane. Usage Report is included as part of Sx Delete Session Response.

Session Report Request and Response Message

Request Message

- On encountering a time or volume threshold limit, user plane generates an Sx Session Report Request message and sends the same to control plane.
- This message contains the Usage report, which indicates the reason for generating the message, specified by Usage Report Trigger.
- In addition to this, the Usage report contains the time or volume measurement.
- If any other URRs are linked to the URR for which the session report request is being generated, then a session report request is generated for those linked URRs as well. For this release, Gy-URRs are not linked with any of the URRs.

Response Message

This message from the Control plane indicates a successful delivery of the Session Report Request message with a cause code. Currently no specific failure handling is done on receiving a failure cause.

Server-Unreachable Support for Gy

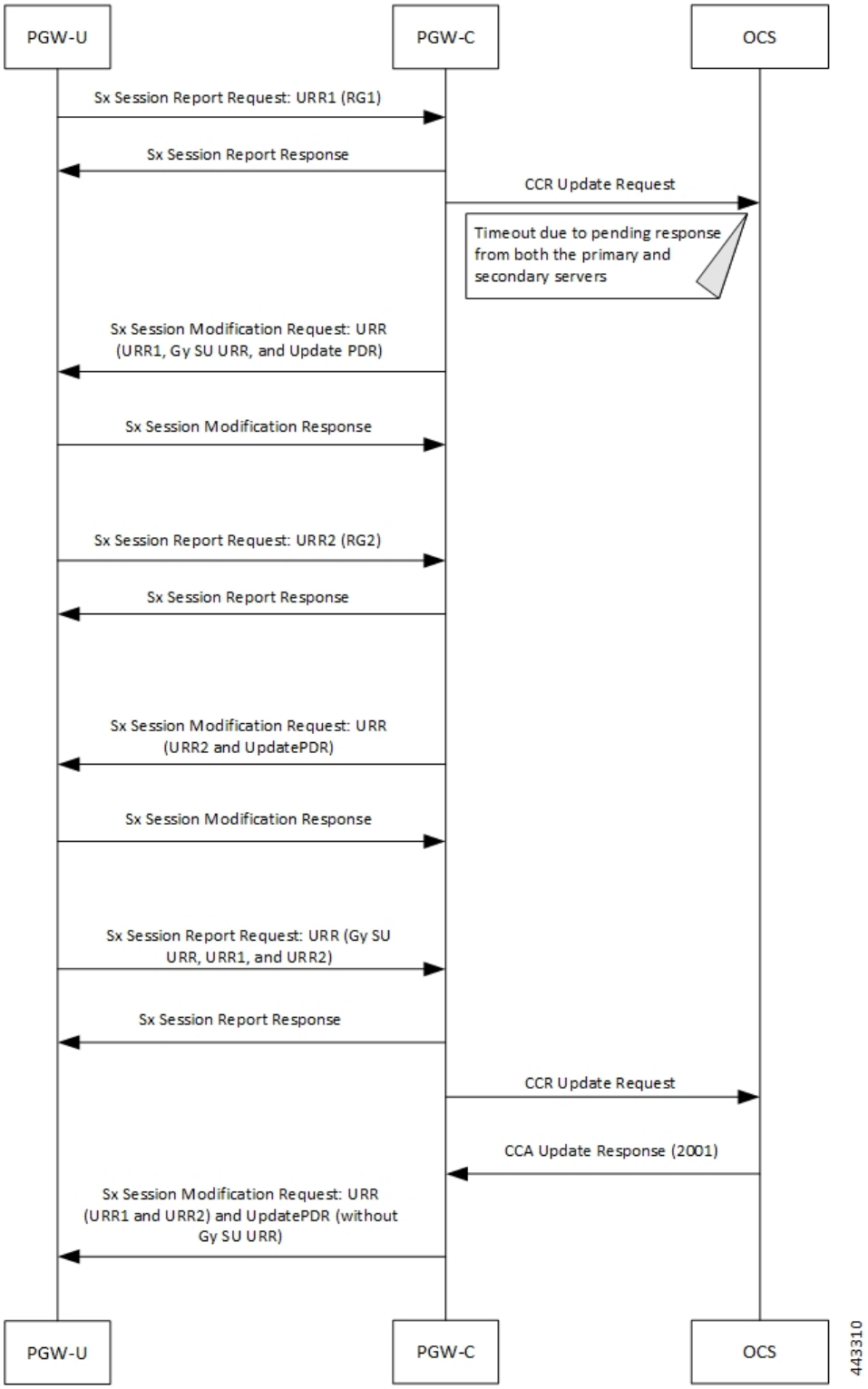
The Server-Unreachable (SU) mechanism is configured on the Control-Plane (CP), for the Gy interface in order to resolve issues that are encountered on the Online Charging System (OCS) or with the connectivity between Policy and Charging Enforcement Function (PCEF) and OCS. The SU configuration provides the options to continue the session even after a failure by providing the option to use configurable interim quota (volume and/or time) and configurable server retries before a session is converted to offline or terminated.

A new Usage Reporting Rule (URR) bucket is created, which contains the SU quota when a Gy session goes into the SU state. The ID for the new URR is generated dynamically when the SU URR is allocated.

In a CUPS User Plane (UP) node, the existing Vector Packet Processor (VPP) streams are modified with a new LC record, which contains the updated SU URR bucket along with the existing set of charging buckets.

When the VPP streams are in the SU state, two quota rows are available, GyURR and SU URR. When GyURR is in the SU state with linked-usage-reporting trigger set, the quota row for the SU URR is linked to the VPP streams.

This section describes the SU call flow in CUPS.


PGW-U
PGW-C
OCS
Sx Session Report Request: URR1 (RG1)
Sx Session Report Response
CCR Update Request
Timeout due to pending response from both the primary and secondary servers
Sx Session Modification Request: URR (URR1, Gy SU URR, and Update PDR)
Sx Session Modification Response
Sx Session Report Request: URR2 (RG2)
Sx Session Report Response
Sx Session Modification Request: URR (URR2 and UpdatePDR)
Sx Session Modification Response
Sx Session Report Request: URR (Gy SU URR, URR1, and URR2)
Sx Session Report Response
CCR Update Request
CCA Update Response (2001)
Sx Session Modification Request: URR (URR1 and URR2) and UpdatePDR (without Gy SU URR)
PGW-U
PGW-C
OCS
443310

Step	Description
1	PGW-U sends a Sx Session Report Request message with URR1 (RG-1) to PGW-C due to internal triggers like Time and Volume or Quota and Threshold.
2	PGW-C acknowledges the request and sends a Sx Session Report Response message to PGW-U.
3	PGW-C sends CCR Update (CCR-U) Request message to both the primary and secondary OCS.
4	When the CCR-U Request messages fail at both primary and secondary OCS, the Gy session enters an SU state. The SU URR is created for the Gy session, and it's linked to the relevant PDRs. PGW-C sends an Sx Session Modification Request message to PGW-U with UpdatePDR, which includes Gy SU URR in the PDR URR list.
5	PGW-U starts updating the usage in both the Gy buckets (URR1 and Gy SU URR1) and sends an Sx Session Modification Response message to PGW-C.
6	PGW-U sends another Sx Session Report Request message with URR2 (RG-2) to PGW-C after the Gy URR bucket exhausts the quota .
7	PGW-C acknowledges the request sends a Sx Session Report Response message to PGW-U.
8	PGW-C sends an Sx Session Modification Request message to PGW-U with UpdatePDR and Update RR2. The UpdatePDR has a modified URR list, which contains both URR2 and Gy SU URR.
9	PGW-U sends an Sx Session Modification Response message to PGW-C.
10	PGW-U sends a Sx Session Report Request message with URR1 (RG-1) and URR2 (RG-2) to PGW-C after the Gy SU URR quota is exhausted.
11	PGW-C acknowledges the request and sends a Sx Session Report Response.
12	PGW-C sends CCR Update (CCR-U) Request message to OCS after an SU retry.
13	OCS sends a CCA Update Response message with the Result-Code as 2001.
14	PGW-C sends an Sx Modification Request message with URR1 (RG-1), URR2 (RG-2), and UpdatePDR (without Gy SU URR) to PGW-U.

New Behavior in CUPS

The new SU mechanism in CUPS, are as follows:

- In a non CUPS architecture, where a single node (P-GW) processes the Gy session state and the data-traffic, an SU URR is created without any messaging delay. However, in the CUPS mode, the CP forms an additional node, which maintains information about the session state and handles any URR requests from the User Plane (UP). Only the CP can associate a Gy session with an SU URR. This messaging between UP and CP causes a delay and the data packets are treated according to the Pending-Traffic-Treatment configuration to complete the communication.
- In a non CUPS architecture, the SU state timer is processed in a different manner compared to the Time-Quota timer. After an SU quota is exhausted, the retry attempt to OCS occurs and a new next-interim-time-quota is started. However, in the CUPS mode, when the SU Time Quota is used and it is reported to CP for the Quota Exhaust, and if the session goes into Server-Unreachable state again, the time elapsed from the last Usage-Report is accounted in the usage.
- It's not recommended to use the **servers-unreachable after-timer-expiry** *timeout_period* CLI command in CUPS. Instead, use the **servers-unreachable after-interim-time** *timeout_period* **server-retries** *retry_count* to achieve similar behavior but with a single retry (set *retry_count* to 1).

Limit-Reached Postprocessing

Limit-reached-post-processing is a non-3GPP, proprietary behavior supported in both CUPS and non-CUPS architecture. This feature allows redirection or restriction operation implemented when the quota is exhausted for the charging-bucket, however, the OCS server is unable to grant the FUI-Redirect or FUI-Restrict. When using this feature, the operator can combine all the rule-matching criteria that are available—for example, to enable IMSI-based matching criteria, and so on—to selectively apply different handling for different subscribers/traffic. Use the following CLI commands to enable the feature.

```
configure
  active-charging service service_name
    rulebase rulebase_name
      post-processing policy always
      end
```

Also, **rule-application post-processing** CLI command must be configured as limit-reached under ACS Ruledef configuration mode.

PTT no-quota Limited Pass

This feature allows the subscriber to use the network while waiting for the response from OCS. The Limited-Pass configuration allows to specify the Volume which the subscriber can consume while waiting for the quota-response from OCS. The usage is accounted in the respective charging bucket and are adjusted against the next-quota allocation.

Use the following CLI commands to enable the feature:

```
configure
  active-charging service service_name
    credit-control
      pending-traffic-treatment noquota limited-pass volume volume
      end
```

Limited Pass Volume is used only for **noquota** case (Rating Group (RG) seeking quota for the first time) and not for **quota-exhausted**. Limited Pass Volume isn't used for subsequent credit requests.

The traffic is allowed to pass until the Limited-Pass Volume gets exhausted. The usage is counted in the respected charging-bucket and adjusted against the "Quota" granted. If the "Quota" allocation is less than the actual usage, immediate reporting towards OCS with the usage-report occurs requesting for more quota allocation. The subsequent incoming packets are handled as per the "quota-exhausted" PTT configuration.

If the Limited Pass Volume is NOT exhausted before the OCS responds with denial of quota, traffic is blocked after the OCS response. The gateway reports usage on Limited-Pass Volume even in for CCR-U (FINAL) (in non-CUPS) or CCR-T (for CUPS) until the OCS responds.

If the Limited Pass Volume is exhausted before the OCS responds, then the subsequent incoming packets for the session are dropped until quota is granted from OCS.

The default pending-traffic-treatment for **noquota** is Drop. The **default pending-traffic-treatment noquota** command removes any Limited Pass Volume size configured.

PTT Quota-Exhausted Limited Pass

Pending-Traffic-Treatment (PTT) Quota-Exhausted Limited-Pass in CUPS architecture is an alternative to the Buffering option. The Buffering option has practical limitations in the high-speed network. Buffering requires packet buffering for large number of packets at the gateway, causing the risk to run out of memory and affecting the bandwidth speed. The PTT Quota-Exhausted Limited Pass allows the traffic to pass through until it reaches the configured limit on the Quota-Exhaust scenarios.

The PTT allows the traffic until the Limited-Pass volume exhausts. The PTT counts and adjusts the usage in the respected charging-bucket against the granted "Quota". If the "Quota" allocation is less than the actual usage, there’s immediate reporting towards OCS with the usage-report and asking for more quota allocation.

If the Limited-Pass Volume doesn’t exhaust before the OCS responds with denial of quota, there’s traffic blockage after the OCS response. Gateway reports the usage in CCR-U (FINAL).

If the Limited-Pass Volume exhausts before the OCS responds, then further incoming packets for the session are dropped until quota is granted from OCS.

The default behavior of pending-traffic-treatment for quota-exhausted is Drop. The default pending-traffic-treatment quota-exhausted CLI command removes any configured Limited-Pass Volume size.

Use the following CLI command to enable the feature:

```
configure
   active-charging service service_name
      credit-control
       pending-traffic-treatment quota-exhausted limited-pass volume volume

         end
```

Note The above CLI command is applicable only in CUPS architecture.

NOTES:

- **limited-pass**: Enables limited access to subscriber when OCS is unreachable.
- **volume** *volume*: Enables limited volume access to subscriber when OCS is unreachable. *volume* specifies the Default Quota size (in bytes) and must be an integer from 1 through 4294967295

Quota-Validity-Time Handling

When the Quota-Validity-Time is received for an MSCC bucket, the same is sent to the User Plane. Since there is no specific IE that can be used directly, the QVT value is filled in the Time-Quota IE, and URR is sent to the User Plane. The lesser QVT or Time-Quota is set in the Time-Quota IE. And, the Usage-Reporting from the User Plane for the Time-Quota trigger, the interpretation is made and the CCR-Update for the Validity-Timeout is generated.

Supported Functionality and Limitations

Basic call flow with Volume-Quota mechanism is supported with the following limitations on P-GW session reporting for Gy interface:

- Only CCR/CCA-I , CCR/CCA-U and CCR/CCA-T, RAR/RAA messages are supported.
- Dynamic Rules with Online Enabled is supported; both at Session-Setup and Mid-Session.
- Predefined Rules (dynamic-only) is supported; both at Session-Setup and Mid-Session. No restriction on configuring the "preemptively request".
- Static-rules with Online Charging are supported.
- Ignore-service-id is supported.
- Volume-Quota/Volume-Threshold mechanisms are supported.
- Event-Triggers (through which the Query URR occurs), and sending of usage information to the OCS is supported.

Important RAT-change functionality is not validated for this release.

- The "updateURR" procedure, through the Sx-Session-Modification procedure where the OCS grants a fresh Quota, is supported.
- Bearer-Level Gy and Subscriber-Level Gy is supported.
- Pending-Traffic-Treatment (PTT) Drop/Pass is supported with following limitations:
  - The scenarios supported for now are no-quota and quota-exhausted.
  - The trigger/re-authorization scenarios are not supported.
  - The PTT action (Forward/Drop) is considered after the quota-get is exhausted.
- Failure scenarios are qualified, which includes:
  - Failure-Handling Terminate, Continue and Retry, and Terminate: With CC-Group/FHT
  - Handling for the Error-Result-Codes (both at MSCC and Command level) is supported.
- Wall-Clock time-quota mechanism is supported.
- Other Time Quota Mechanisms (Discrete Time Period and Continuous-Time-Period) are not supported.
- Final-Unit-Indication Terminate mechanism is supported.
- FUI-Restrict is not supported.

- Mid-Session Rule Installation/Removal/Modification is supported.
- RAR mechanism is supported.
- Server-Unreachable (SU) mechanism is now supported with minor change in behavior compared to non-CUPS P-GW.
  - When an URR needs quota at UP, the usage-report is generated to CP and until the CP responds with the linked SU_URR, the packets matching this URR are treated with Pending-Traffic-Treatment configuration.
  - When the SU Time Quota is used and it's reported to CP for the Quota Exhaust, and if the session goes into Server-Unreachable state again, the time elapsed from the last Usage-Report is accounted in the usage.
- Pending-Traffic-Treatment Buffer mechanism is not supported.
- The “send-ccri on traffic-start” is supported.
- Quota-Hold-Time is supported.
- Quota-Consumption-Time mechanism is not supported.
- Quota-Validity-Time is supported.
- Triggering Gz records from Gy, when any event in Gy occurs, is supported; Gy-Gz sync is not supported.
- Triggering Rf records from Gy, when any event in Gy occurs, is not supported.
- Configuring different "rating-group" value other than the "content-id" is supported.
  - The RG 0 is not supported.
- Trigger to PCRF for the Out-of-Credit, Reallocation-of-Credit events are not qualified.

Important Event-trigger Out-of-Credit towards PCRF is validated with a limitation of having only one time Grant-Quota (Keeping Total Volume and Granted Volume at same value).

- The delayed response from OCS for the CCR-I is supported.
- Service-Specific-Units are not supported.
- Tariff-Time change is supported as per 3GPP specification.
- Quota-Retry Timer is supported.
- The **diameter mscc-final-unit-action terminate session** CLI command under Credit Control Configuration mode is supported.
- FUI-Redirect is supported with following limitations:
  - Redirection for HTTPs is not supported.
  - The FUI-Redirect with Filter-IDs/Filter-Rules are not supported.
  - The WSP Protocol is not supported.

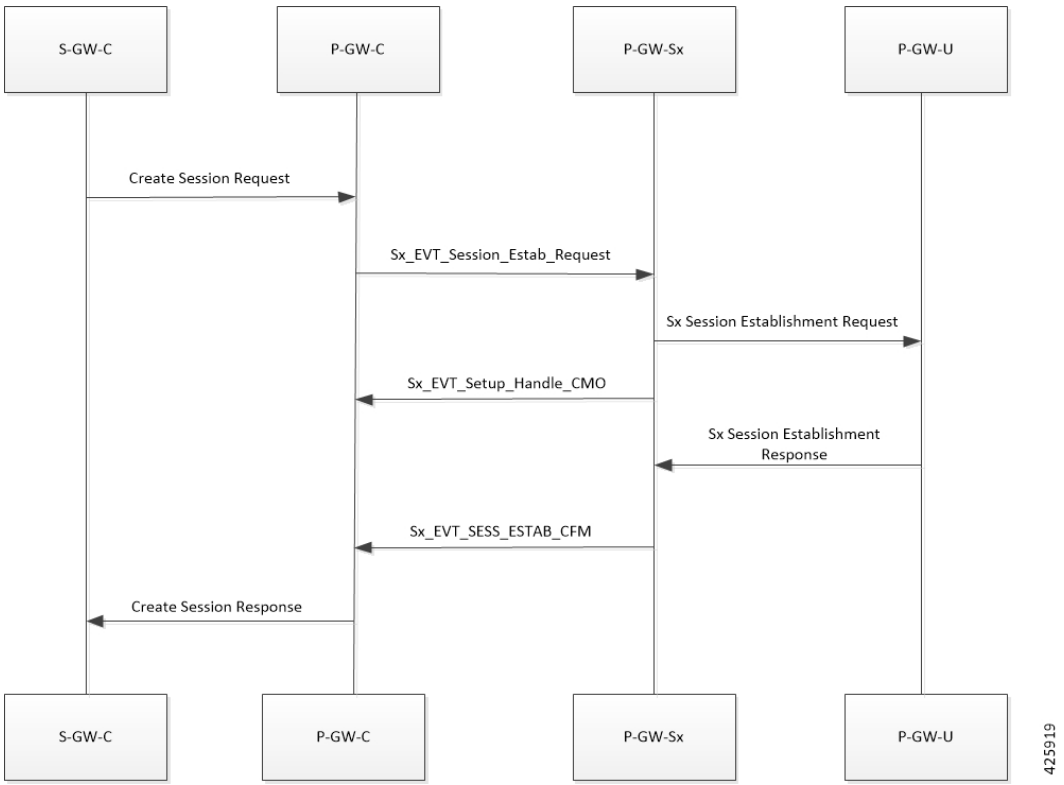
- In accordance with 3GPP specification, the Redirected-Traffic also gets redirected if it hits the rule that is in FUI-Redirect. There is no provision to allow the redirected-traffic to pass through.
  - In accordance with 3GPP specification, the CUPS architecture adheres to **no diameter fui-redirected-flow allow** CLI command behavior.
- The **redirect-require-user-agent** CLI command is not supported; the redirection continues to work even if the user-agent is not present.
- Appending the original URL is not supported.
- The **diameter redirect-validity-timer immediate** CLI command is supported. However, **diameter redirect-validity-timer traffic-start** CLI command is not supported.
- Token based mechanism, to come out of Redirection, is not supported. To end the redirection in CUPS, OCS sends Redirect Validity-Time or RAR.
- FUI-Redirection is supported only for the URL, similar to the behavior in non-CUPS architecture.

- Rulebase change from PCRF/OCS is supported.

P-GW Session Reporting with Gz Interface

This section describes P-GW session reporting with Gz interface.

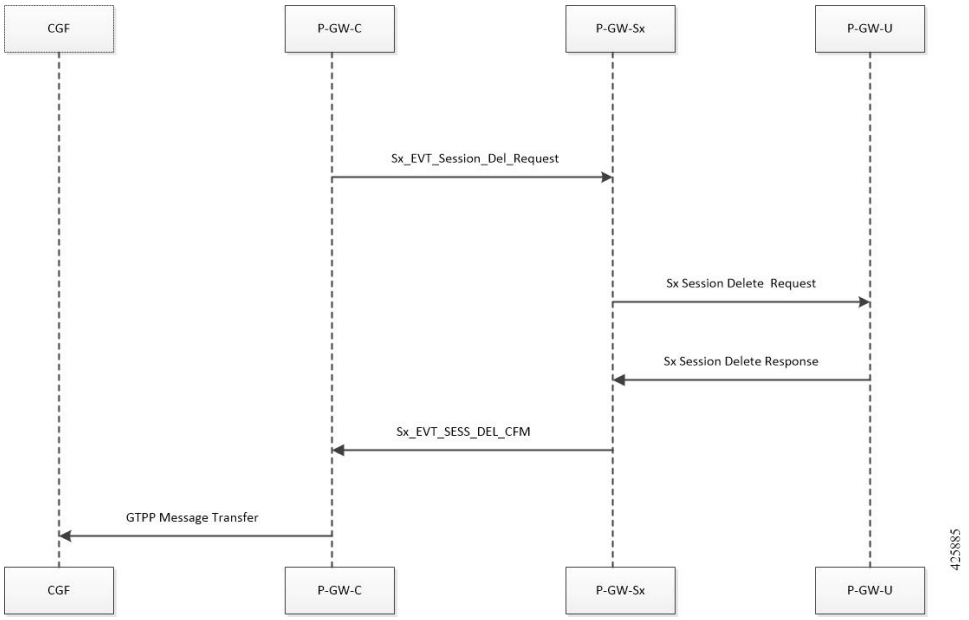
URR Support in Session Establishment Request



- User plane module supports the storage of a list of URRs received as part of session establishment request.

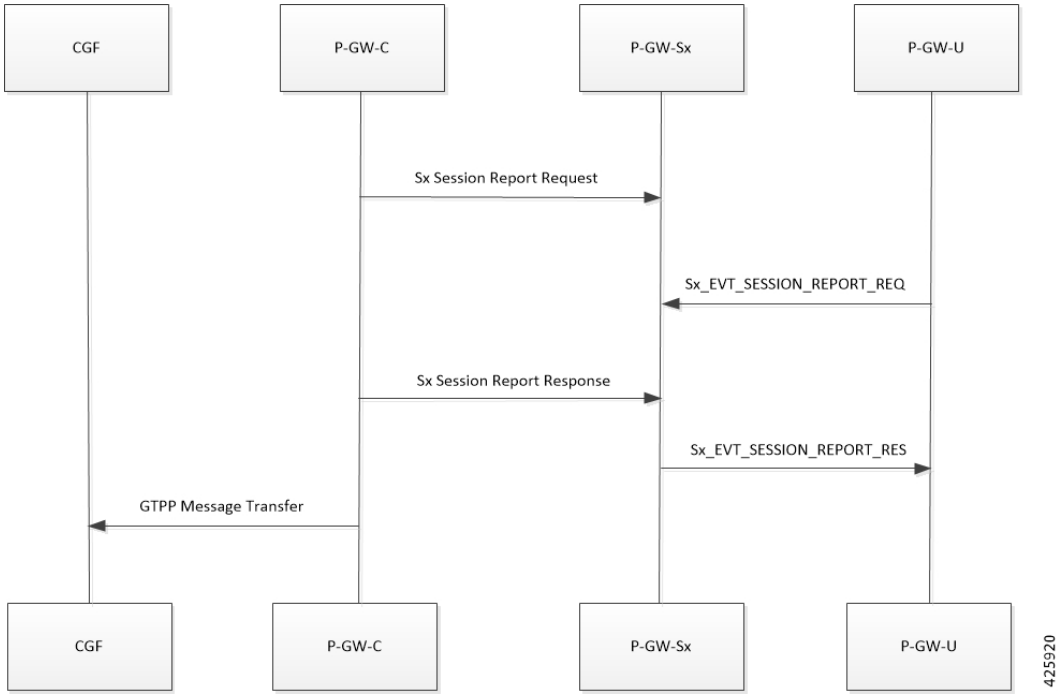
- Each PDR is associated with one or more URRs.
- A particular URR is linked to another URR.
- Each URR contains the measurement method (time or volume), and reporting triggers that indicates the event on which the user plane has to send usage report.

Session Delete Response



This message sent from the User Plane is in response to a session deletion request from control plane. This results in the termination of the Sx session at User plane. Usage Report is included as part of SX Delete Session Response.

Session Report Request and Response Message



Request Message

- On encountering a time or volume threshold limit, user plane generates an Sx Session Report Request message and sends the same to control plane.
- This message contains the Usage report, which indicates the reason for generating the message, specified by Usage Report Trigger.
- In addition to this, the Usage report contains the time or volume measurement.
- If any other URRs are linked to the URR for which the session report request is being generated, then a session report request is generated for those linked URRs as well.

Response Message

This message from the Control plane indicates a successful delivery of the Session Report Request message with a cause code. Currently no specific failure handling is done on receiving a failure cause.

Bit Rate Mapping Support

P-GW converts the bit rate value that it receives from PCRF from bps to kbps. This conversion may lead to truncation of fractional value to nearest integer (floor) value and lead to loss of information. 3GPP suggested that if the conversion from bps to kbps leads to a fractional value, then it should be rounded up to the nearest integer value (ceil) value and sent to the access side.

Note Design changes are done to ensure rounded down (floor) value from bps to kbps is sent on the PFCP interface.

Standards Compliance

The bit rate mapping feature complies with 3GPP TS 29.274 release 12.

Configuring the Bit Rate Mapping Feature

To configure the rounded up (ceil) value for bit rate from bps to kbps in APN-AMBR, GBR, and MBR on P-GW, perform the following steps:

```
configure
  context context_name
    pgw-service service_name
      [ no ] egtp bitrates-rounded-down-kbps
      end
```

To configure the rounded down (floor) value for bit rate from bps to kbps in APN-AMBR, GBR, and MBR on P-GW, perform the following steps:

```
configure
  context context_name
    pgw-service service_name
      egtp bitrates-rounded-down-kbps
      end
```

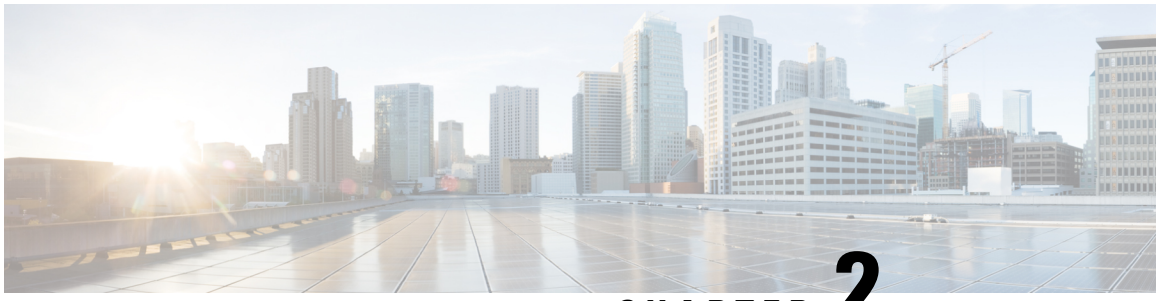
New Behavior in CUPS

By default, the rounded up value of bit rate in kbps for APN-AMBR, MBR, and GBR will be sent on the Sx and GTP interfaces. To enable the rounding down behavior, CLI must be configured.

Standards Compliance

The User Plane in CUPS complies with the following standards:

- 3GPP specification 23.214 release 14.0: Universal Mobile Telecommunications System (UMTS); LTE; Architecture enhancements for control and user plane separation of EPC nodes
- 3GPP specification 29.244 release 14.0: LTE; Interface between the Control Plane and the User Plane of EPC Nodes
- 3GPP specification 23.401 release 14.0: 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access



CHAPTER 2

Configuring User Plane in CUPS

This section describes the CLI commands available to configure User Plane in CUPS.

Important For information related to following configurations, refer the *Ultra Packet Core CUPS Sx Interface Administration and Reference Guide*:

- *Configuring Sx Service for CUPS*
- *Configuring Sx-u Interface for CUPS*
- *Configuring Sx Demux for CUPS*

Important

- The following configuration limit applies in CUPS:
  - Rulebase - 512
  - Ruledef - 2500
  - Charging-action - 2048
- The following CLI command is not recommended to be used, with active subscriber sessions, in production environment: **no active-charging service** *service_name*

- Configuring User Plane Service, on page 37
- Associating GTP-U Service with User Plane Service, on page 38
- Associating Sx Service to User Plane Service, on page 39
- Recommended Timers, on page 39

Configuring User Plane Service

Use the following CLI commands to configure the User Plane service.

```
configure
  context context_name
```

```
[ no ] user-plane-service service_name
end
```

NOTES:

- **user-plane-service** *service_name*: Creates the specified User Plane service name to allow configuration of User Plane service. The *service_name* is a mandatory parameter to define the User Plane service.
- **[no] user-plane-service** *service_name*: Removes the User Plane service from the particular context.
- By default, the CLI is disabled.

Starting a User Plane Service

The following minimum and critical parameters must be configured to start the User Plane service:

- One Sx-Service.
- Three GTP-U Services of interface type P-GW ingress, S-GW-ingress, and S-GW-egress.

Important Removal or change of any critical parameters from User Plane service results in the User Plane service getting stopped.

The services that are associated with User Plane service should be in running mode. Else, stop in any associated service triggers stopping of User Plane service.

Associating GTP-U Service with User Plane Service

To associate the GTPU service with the User Plane service, execute the following CLI commands:

```
configure
  context context_name
    user-plane-service service_name
    [ no ] associate gtpu-service gtpu_service_name { pgw-ingress |
sgw-ingress | sgw-egress }
    end
```

NOTES:

- **no**: Removes association of GTP-U service with the specified interface type from User Plane service.
- **associate**: Associates User Plane service with GTP-U service.
- **gtpu-service** *gtpu_service_name*: Specifies the GTP-U service for the User Plane service.
- **pgw-ingress**: Configures the interface type as P-GW ingress.
- **sgw-ingress**: Configures the interface type as S-GW ingress.
- **sgw-egress**: Configures the interface type as S-GW egress.
- By default, this command is disabled.

Associating Sx Service to User Plane Service

Use the following CLI commands to associate Sx service with User Plane service.

```
configure
  context context_name
    user-plane-service service_name
      associate sx-service sx_service_name
      no associate sx-service
      end
```

NOTES:

- **no** : Removes association of Sx service from User Plane service.
- Associating Sx service with User Plane service is a mandatory parameter.
- By default, this CLI command is disabled.

Recommended Timers

The following table provides the recommended timer values for CLI commands related to IPSec, Sx, and SRP.

IPSEC	**CP**	**UP**
ikev2-ikesa max-retransmission	*3*	*3*
ikev2-ikesa retransmission-timeout	*1000*	*1000*
keepalive	**interval** *4* **timeout** *1* **num-retry** *4*	**interval** *5* **timeout** *2* **num-retry** *4*
Sx	**CP**	**UP**
sx-protocol heartbeat interval	*10*	*10*
sx-protocol heartbeat retransmission-timeout	*5*	*5*
sx-protocol heartbeat max-retransmissions	*4*	*4*
sxa max-retransmissions	*4*	*4*
sxa retransmission-timeout-ms	*5000*	*5000*
sxb max-retransmissions	*4*	*4*
sxb retransmission-timeout-ms	*5000*	*5000*
sxab max-retransmissions	*4*	*4*
sxab retransmission-timeout-ms	*5000*	*5000*

IPSEC	CP	UP
sx-protocol association reattempt-timeout	*60*	*60*
SRP	**CP**	**UP**
hello-interval	*3*	*3*
dead-interval	*15*	*15*

Recommended Configurations

Following are the recommended configurations and restrictions related to Sx and SRP over IPSec:

- The multihop BFD timer between CP and UP must be seven seconds (for Data UPs).
- The singlehop BFD must be enabled on all the contexts (CP GW/Billing and UP Gn/Gi).
- Inter-chassis multihop BFD must be enabled for CP-CP ICSR and UP-UP ICSR (IMS UP).
- The SRP-IPSec ACL must be configured for TCP protocol instead of IP protocol.
- The Sx-IPSec ACL must be configured for UDP protocol instead of IP protocol.

Example Configurations in CP

Multihop BFD Configuration VPC-DI

The following is an example of multihop BFD configuration with seven seconds timer.

```
bfd-protocol
      bfd multihop-peer 209.165.200.226 interval 350 min_rx 350 multiplier 20
      bfd multihop-peer 209.165.200.227 interval 350 min_rx 350 multiplier 20
      bfd multihop-peer 209.165.200.225 interval 350 min_rx 350 multiplier 20
      bfd multihop-peer 209.165.200.230 interval 350 min_rx 350 multiplier 20
      bfd multihop-peer 209.165.200.228 interval 350 min_rx 350 multiplier 20
      bfd multihop-peer 209.165.200.229 interval 350 min_rx 350 multiplier 20
    #exit
```

Multihop BFD Configuration VPC-SI

The following is an example of multihop BFD configuration with three seconds timer.

```
bfd-protocol
      bfd multihop-peer 209.165.200.226 interval 150 min_rx 150 multiplier 20
      bfd multihop-peer 209.165.200.227 interval 150 min_rx 150 multiplier 20
      bfd multihop-peer 209.165.200.225 interval 150 min_rx 150 multiplier 20
      bfd multihop-peer 209.165.200.230 interval 150 min_rx 150 multiplier 20
      bfd multihop-peer 209.165.200.228 interval 150 min_rx 150 multiplier 20
      bfd multihop-peer 209.165.200.229 interval 150 min_rx 150 multiplier 20
    #exit
```

BGP Configuration

The following is an example of BGP configuration with recommended timers.

```
router bgp 1111
      router-id 209.165.200.225
      maximum-paths ebgp 15
```

```
      neighbor 209.165.200.250 remote-as 1000
      neighbor 209.165.200.250 ebgp-multihop
      neighbor 209.165.200.250 update-source 209.165.200.225
      neighbor 1111:2222::101 remote-as 1000
      neighbor 1111:2222::101 ebgp-multihop
      neighbor 1111:2222::101 update-source 1111:2222::1
      bgp graceful-restart restart-time 120
      bgp graceful-restart stalepath-time 300
      timers bgp keepalive-interval 30 holdtime-interval 90 min-peer-holdtime-interval 0
server-sock-open-delay-period 10
      address-family ipv4
        redistribute connected
      #exit
      address-family ipv6
        neighbor 1111:2222::101 activate
        redistribute connected
      #exit
    #exit
```

Singlehop BFD Configuration

The following is an example of singlehop BFD configuration with three seconds timer.

```
interface bgp-sw1-2161-10
      ip address 209.165.200.233 209.165.200.255
      ipv6 address 1111:222::9/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-11
      ip address 209.165.200.234 209.165.200.255
      ipv6 address 1111:222::10/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-12
      ip address 209.165.200.235 209.165.200.255
      ipv6 address 1111:222::11/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-3
      ip address 209.165.200.226 209.165.200.255
      ipv6 address 1111:222::2/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-4
      ip address 209.165.200.227 209.165.200.255
      ipv6 address 1111:222::3/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-5
      ip address 209.165.200.228 209.165.200.255
      ipv6 address 1111:222::4/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-6
      ip address 209.165.200.229 209.165.200.255
      ipv6 address 1111:222::5/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-7
      ip address 209.165.200.230 209.165.200.255
      ipv6 address 1111:222::6/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-8
```

```
      ip address 209.165.200.231 209.165.200.255
      ipv6 address 1111:222::7/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
    interface bgp-sw1-2161-9
      ip address 209.165.200.232 209.165.200.255
      ipv6 address 1111:222::8/112 secondary
      bfd interval 999 min_rx 999 multiplier 3
    #exit
```

Static Route for Multihop BFD Configuration

The following is an example of static route multihop BFD configuration.

```
ip route static multihop bfd UP-5 209.165.200.240 209.165.200.245
    ip route static multihop bfd UP-6 209.165.200.240 209.165.200.246
    ip route static multihop bfd UP-9 209.165.200.240 209.165.200.247
    ip route static multihop bfd UP-10 209.165.200.240 209.165.200.248
    ip route static multihop bfd UP-7 209.165.200.240 209.165.200.249
    ip route static multihop bfd UP-8 209.165.200.240 209.165.200.250
```

Static Route for Singlehop BFD Configuration

The following is an example of static route singlehop BFD configuration.

```
ip route static bfd bgp-sw1-2161-3 209.165.200.230
    ip route static bfd bgp-sw1-2161-4 209.165.200.230
    ip route static bfd bgp-sw1-2161-5 209.165.200.230
    ip route static bfd bgp-sw1-2161-6 209.165.200.230
    ip route static bfd bgp-sw1-2161-7 209.165.200.230
    ip route static bfd bgp-sw1-2161-8 209.165.200.230
    ip route static bfd bgp-sw1-2161-9 209.165.200.230
    ip route static bfd bgp-sw1-2161-10 209.165.200.230
    ip route static bfd bgp-sw1-2161-11 209.165.200.230
    ip route static bfd bgp-sw1-2161-12 209.165.200.230
```

IPSec ACL Configuration

The following is an example IPSec ACL configuration in CP.

```
ip access-list UP-1
      permit udp host 209.165.200.225 host 209.165.200.226
    #exit
```

IPSec Transform Set Configuration

The following is an example of IPSec Transform Set configuration in CP.

```
ikev2-ikesa transform-set ikesa-UP-1
      encryption aes-cbc-256
      group 14
      hmac sha2-256-128
      lifetime 28800
      prf sha2-256

    ipsec transform-set A-UP-1
      encryption aes-cbc-256
      hmac sha2-256-128
      group 14
```

IPSec Crypto Map Configuration

The following is an example of IPSec Crypto Map configuration in CP.

```
crypto map UP-1 ikev2-ipv4
      match address UP-1
      authentication local pre-shared-key encrypted key secretkey
      authentication remote pre-shared-key encrypted key secretkey
      ikev2-ikesa max-retransmission 3
      ikev2-ikesa retransmission-timeout 1000
      ikev2-ikesa transform-set list ikesa-UP-1
      ikev2-ikesa rekey
      keepalive interval 4 timeout 1 num-retry 4
      control-dont-fragment clear-bit
      payload foo-sa0 match ipv4
        ipsec transform-set list A-UP-1
        lifetime 300
        rekey keepalive
      #exit
      peer 192.1.1.1
      ikev2-ikesa policy error-notification
    #exit
```

Sx Configuration

The following is an example of Sx configuration in CP.

```
sx-service SX-1
      instance-type controlplane
      sxa max-retransmissions 4
      sxa retransmission-timeout-ms 5000
      sxb max-retransmissions 4
      sxb retransmission-timeout-ms 5000
      sxab max-retransmissions 4
      sxab retransmission-timeout-ms 5000
      n4 max-retransmissions 4
      n4 retransmission-timeout-ms 5000
      sx-protocol heartbeat interval 10
      sx-protocol heartbeat retransmission-timeout 5
      sx-protocol heartbeat max-retransmissions 4
      sx-protocol compression
      sx-protocol supported-features load-control
      sx-protocol supported-features overload-control
    exit
end
```

Example Router Configurations

Static Routes for Interface

The following is an example configuration of static route for interface.

```
ip route 209.165.200.224/27 Vlan1111 209.165.200.225
ip route 209.165.200.224/27 Vlan1111 209.165.200.226
ip route 209.165.200.224/27 Vlan1111 209.165.200.227
ip route 209.165.200.224/27 Vlan1111 209.165.200.228
ip route 209.165.200.224/27 Vlan1111 209.165.200.229
ip route 209.165.200.224/27 Vlan1111 209.165.200.230
ip route 209.165.200.224/27 Vlan1111 209.165.200.231
ip route 209.165.200.224/27 Vlan1111 209.165.200.232
ip route 209.165.200.224/27 Vlan1111 209.165.200.233
ip route 209.165.200.224/27 Vlan1111 209.165.200.234
```

Static Routes for Singlehop BFD

The following is an example configuration of static route for singlehop BFD.

```
ip route static bfd Vlan1111 209.165.200.225
ip route static bfd Vlan1111 209.165.200.226
ip route static bfd Vlan1111 209.165.200.227
ip route static bfd Vlan1111 209.165.200.228
ip route static bfd Vlan1111 209.165.200.229
ip route static bfd Vlan1111 209.165.200.230
ip route static bfd Vlan1111 209.165.200.231
ip route static bfd Vlan1111 209.165.200.232
ip route static bfd Vlan1111 209.165.200.233
ip route static bfd Vlan1111 209.165.200.234
```

Interface for Singlehop BFD

The following is an example configuration of interface for singlehop BFD.

```
interface Vlan1111
  no shutdown
  bandwidth 10000000
  bfd interval 999 min_rx 999 multiplier 3
  no bfd echo
  ip address 209.165.200.224/27
  ipv6 address 1111:222::1/112
```

BGP Configuration

The following is an example of BGP configuration with recommended timers.

```
router bgp 1000
  router-id 209.165.200.226
  timers bgp 30 90
  timers bestpath-limit 300
  timers prefix-peer-timeout 30
  timers prefix-peer-wait 90
  graceful-restart
  graceful-restart restart-time 120
  graceful-restart stalepath-time 300
```

Example Configurations in UP

IPSec ACL Configuration

The following is an example of IPSec ACL configuration in UP.

```
ip access-list CP-1
      permit udp host 209.165.200.225 host 209.165.200.226
    #exit
```

IPSec Transform Set Configuration

The following is an example of IPSec Transform Set configuration in UP.

```
ipsec transform-set A-CP-1
      encryption aes-cbc-256
      hmac sha2-256-128
      group 14

    ikev2-ikesa transform-set ikesa-CP-1
      encryption aes-cbc-256
      group 14
      hmac sha2-256-128
      lifetime 28800
      prf sha2-256
```

IPSec Crypto Map Configuration

The following is an example of IPSec Crypto Map configuration in UP.

```
crypto map CP-1 ikev2-ipv4
      match address CP-1
      authentication local pre-shared-key encrypted key secretkey
      authentication remote pre-shared-key encrypted key secretkey
      ikev2-ikesa max-retransmission 3
      ikev2-ikesa retransmission-timeout 1000
      ikev2-ikesa transform-set list ikesa-CP-1
      ikev2-ikesa rekey
      keepalive interval 5 timeout 2 num-retry 4
      control-dont-fragment clear-bit
      payload foo-sa0 match ipv4
        ipsec transform-set list A-CP-1
      #exit
      peer 209.165.200.230
      ikev2-ikesa policy error-notification
    #exit
```

Sx Configuration

The following is an example of Sx configuration in UP.

```
sx-service SX-1
      instance-type userplane
      sxa max-retransmissions 4
      sxa retransmission-timeout-ms 5000
      sxb max-retransmissions 4
      sxb retransmission-timeout-ms 5000
      sxab max-retransmissions 4
      sxab retransmission-timeout-ms 5000
      n4 max-retransmissions 4
      n4 retransmission-timeout-ms 5000
      sx-protocol heartbeat interval 10
      sx-protocol heartbeat retransmission-timeout 5
      sx-protocol heartbeat max-retransmissions 4
      sx-protocol compression
    exit
```

Example SRP Configurations

IPSec ACL Configuration

The following is an example of IPSec ACL configuration for SRP.

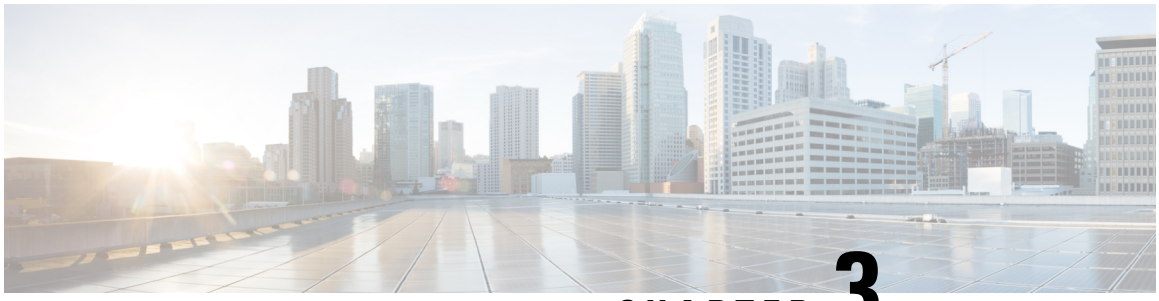
```
ip access-list SRP
      permit tcp host 209.165.200.227 host 209.165.200.228
    #exit
```

SRP Configuration

The following is an example of SRP configuration.

```
configure
  context srp
    bfd-protocol
      bfd multihop-peer 209.165.200.225 interval 999 min_rx 999 multiplier 3
    #exit
configure
  context srp
    service-redundancy-protocol
      chassis-mode primary
```

```
      hello-interval 3
      dead-interval 15
      monitor bfd context srp 209.165.200.226 chassis-to-chassis
      monitor bgp context gi-pgw 209.165.200.245
      monitor bgp context gi-pgw 3333:888::1
      monitor bgp context saegw 209.165.200.245
      monitor bgp context saegw 3333:888::2
      peer-ip-address 209.165.200.227
      bind address 209.165.200.228
    #exit
    ip route static multihop bfd srp 209.165.200.229 209.165.200.245
    ip route 209.165.201.1 209.165.202.129 209.165.200.230 SRP-Physical-2102
    ip route 209.165.201.2 209.165.202.130 209.165.200.231 SRP-Physical-2102
    ip route 209.165.201.3 209.165.202.131 209.165.200.232 SRP-Physical-2102
    ip igmp profile default
    #exit
  #exit
end
```



CHAPTER 3

Monitoring and Troubleshooting User Plane in CUPS

This section provides information about the CLI commands available to monitor and/or troubleshoot User Plane in CUPS.

- Monitoring and Troubleshooting User Plane in CUPS, on page 47
- SNMP Traps, on page 47
- Show Commands, on page 48

Monitoring and Troubleshooting User Plane in CUPS

This section provides information about the CLI commands available to monitor and/or troubleshoot User Plane in CUPS.

SNMP Traps

The following traps are available after session recovery in the User Plane node:

- **starManagerFailure**: This trap is generated when there is failure in the Software manager.
- **starTaskFailed**: This trap is generated when a noncritical task has failed and the appropriate recovery steps begin.
- **starTaskRestart**: This trap is generated when a noncritical task has restarted after an earlier failure.
- **starSessMgrRecoveryComplete**: This trap is generated when Session Manager recovery completes. This is typically caused by the failure of Session Manager task and successful completion of recovery.
- **starManagerRestart**: This trap is generated when the identified manager task has been restarted.

Show Commands

show configuration

This command displays the following fields:

```
saegw-service
associate sgw-service
associate pgw-service
associate gtpu-service up-tunnel
associate sx-service
```

show-gtpu-statistics

Executing this show command displays the following output:

- Session Stats:
  - Current
  - Current (IMS-media)
  - Total Setup
  - Total Setup (IMS-media)
  - Current gtpu v0 sessions
  - Current gtpu v1 sessions
- Total Data Stats:
  - Uplink Packets
  - Uplink Bytes
  - Downlink Packets
  - Downlink Bytes
  - Packets Discarded
  - Bytes Discarded
  - Uplink Packets (IMS-media
  - Uplink Bytes (IMS-media)
  - Downlink Packets (IMS-media
  - Downlink Bytes (IMS-media)
  - Packets Discarded (IMS-media
  - Bytes Discarded (IMS-media
- QoS Stats:

- QCI <n>:
  - Uplink Packets
  - Uplink Bytes
  - Downlink Packets
  - Downlink Byte
  - Packets Discarded
  - Bytes Discarded
- Non-Std QCI(Non-GBR):
  - Uplink Packets
  - Uplink Bytes
  - Downlink Packets
  - Downlink Byte
  - Packets Discarded
  - Bytes Discarded

- Non-Std QCI(GBR):
  - Uplink Packets
  - Uplink Bytes
  - Downlink Packets
  - Downlink Byte
  - Packets Discarded
  - Bytes Discarded
- Total uplink packets GBR QCI's:
  - Total uplink Bytes GBR QCI's
  - Total Downlink packets GBR QCI's
  - Total Downlink Bytes GBR QCI's
  - Total uplink packets Non-GBR QCI's
  - Total uplink Bytes Non-GBR QCI's
  - Total Downlink packets Non-GBR QCI's
  - Total Downlink Bytes Non-GBR QCI's
- Path Management Messages:
  - Echo Request Rx

- Echo Response Rx
- Echo Request Tx
- Echo Response Tx
- SuppExtnHdr Tx
- SuppExtnHdr Rx

- Peer Stats:
  - Total GTPU Peers
  - Total GTPU Peers with Stats

- Tunnel Management Messages:
  - Error Indication Tx
  - Error Indication Rx
  - Error Indication Rx Discarded

- Optimization Stats:
  - Total Packets Input
  - Total Packets Optimized
  - Total TCP Packets Input:
  - Total TCP Packets Optimized:
  - Total UDP Packets Input
  - Total UDP Packets Optimized
  - Total Fragments Input

- IPSec Data Stats:
  - Discards Due To IPSec Tunnel Not Present
    - Packets Discarded
    - Bytes Discarded
    - Err-Ind Tx Discarded

Note In CUPS, the "Packets Discarded" statistics are the aggregate of packets dropped at the Session manager and packets dropped at VPP. As VPP handles majority packets, the packet drops at VPP can only be categorized broadly under these statistics.

You can view specific packet drop reasons only for packets dropped at session manager. Packets dropped at VPP are categorized under Packets Discarded counter in the **show gtpu statistics** CLI.

show module p2p user-plane-ipv6-addr

Executing this show command displays the following output:

- Control-Plane Sx-Service name
  - Priority
  - User-Plane ip
  - version
  - update/rollback time

show saegw-service all

The output of this command has been enhanced to include the following new field in support of the Sx Service associated with an SAEGW Service.

sx-service

show saegw-service name

The output of this command is similar to the **show saegw-service all** CLI command and displays the field for the specified saegw-service name.

show service all

The output of this command has been modified to include user-plane service and its related parameters.

- Context ID
- Service ID
- Context Name
- Service Name
- State
- MaxSessions
- Type

show subscriber all

The output of this command has been modified to include user-plane service and its related parameters:

- Access type
  - user-plane
- Access Tech

- Call State
- Access CSCF Status
- Link Status
- Network Type
- CALLID
- MSID
- USERNAME
- IP
- TIME-IDLE

show subscribers user-plane-only all

Executing this show command displays the following output:

- Access Type
- Interface Type
- Call State
- CALL ID
- LOCAL SEID
- IP
- PDN-INSTANCE
- TIME-IDLE

show subscribers user-plane-only called/seid *called/seid* flow flow-id *flow-id*

Executing this show command displays the following output:

- Callid
  - Interface Type
  - IP address
  - Flow ID
  - Uplink pkts
  - Downlink pkts
  - Uplink bytes
  - Downlink bytes
  - UE IP address

    - UE Port
    - Server IP address
    - Server Port
    - Protocol
- Total Flows found
- Total subscribers matching specified criteria

show subscribers user-plane-only called/seid *called/seid* flows full

Executing this show command displays the following output:

- Callid
    - Interface Type
    - IP address
    - Flow ID
    - Uplink pkts
    - Downlink pkts
    - Uplink bytes
    - Downlink bytes
    - UE IP address
    - UE Port
    - Server IP address
    - Server Port
    - Protocol
    - Flow ID
    - Uplink pkts
    - Downlink pkts
    - UE IP address
    - UE Port
    - Server IP address
    - Server Port
    - Protocol
- Total Flows found

- Total subscribers matching specified criteria

show subscribers user-plane-only called/seid *called/seid* flows

Executing this show command displays the following output:

- Sessmgr Instance
  - Application Protocol
  - Transport Protocol
    - Tethered Flow
      - Recovered Flow
  - Total Number of Active flows

show subscribers user-plane-only callid *call_id* pdr all

Executing this show command displays the following output:

- Source Interface
- Type
- Destination Interface
- Type
- vv
- PDR-ID
- Linked FAR-ID
- Linked URR-ID
- Linked QER-ID
- Total subscribers matching specified criteria

show subscribers user-plane-only callid/seid *callid/seid* pdr full all

Executing this show command displays the following output:

- Callid
  - Interface Type
  - IP address
- PDR-ID
- Hits

- Match Bypassed
- Matched Bytes
    - Precedence
    - Source Interface
- Matched Packets
- SDF Filter(s)
    - Filter 1
        - Protocol
        - Src IP Addr
        - Src Port
        - Dst IP Addr
        - Dst Port
- SPI
    - Local F-TEID
    - Outer header removal
    - Application ID
- Linked FARID
    - Destination Interface
    - Apply Action
    - Outer Header Creation
    - Remote TEID
    - Remote IP Address
    - Remote Port
- Linked QERID
    - PDR-ID
    - Hits
    - Match Bypassed
    - Matched Bytes
        - Precedence
        - Source Interface

- SDF Filter(s)
  - Filter 1
    - Protocol
    - Src IP Addr
    - Src Port
    - Dst IP Addr
    - Dst Port
    - SPI
- Local F-TEID
- Outer header removal
- Application ID

- Linked FARID
  - Destination Interface
  - Apply Action
  - Outer Header Creation
  - Remote TEID
  - Remote IP Address
  - Remote Port

- Total PDRs found
- Total subscribers matching specified criteria

show subscribers user-plane-only callid/seid *callid/seid* pdr id *pdr-id*

Executing this show command displays the following output:

- Callid
  - Interface Type
  - IP address

- PDR-ID
- Hits
- Match Bypassed
- Matched Bytes
  - Precedence

- Source Interface

- Matched Packets
- SDF Filter(s)
  - Filter 1
    - Protocol
    - Src IP Addr
    - Src Port
    - Dst IP Addr
    - Dst Port
    - SPI

- Local F-TEID
- Outer header removal
- Application ID
- Linked FARID
  - Destination Interface
  - Apply Action
  - Outer Header Creation
  - Remote TEID

- Remote IP Address
- Remote Port
- Linked QERID
- Total PDRs found
- Total subscribers matching specified criteria

show subscribers user-plane-only flows

Executing this show command displays the following output:

- Sessmgr Instance
  - Application Protocol
  - Transport Protocol
    - Tethered Flow
      - Recovered Flow

- Flow-ID
- Bytes-Up
- Bytes-Down
- Pkts-Up
- Total Number of Active flows
- Total subscribers matching specified criteria

show subscribers user-plane-only full all

Executing this show command displays the following output:

- Local SEID
- Remote SEID
- State
- Connect Time
- Idle time
- Access Type
- Network Type
- user-plane-service-name
- Callid
- Interface Type
- Card/Cpu
- IP allocation type
- IP address
- Source context
- Destination context
- PDN-Instance
- User-plane-Sx-addr
- Control-plane-Sx-addr
- Number of associated PDRs
- Number of associated FARs
- Number of associated QERs
- Number of associated URRs
- input pkts

- output pkts
- input bytes
- output bytes
- input bytes dropped
- output bytes dropped
- input pkts dropped
- output pkts dropped
- pk rate from user(bps)
- pk rate to user(bps)
- ave rate from user(bps)
- ave rate to user(bps)
- sust rate from user(bps)
- sust rate to user(pps)
- pk rate from user(pps)
- pk rate to user(pps)
- ave rate from user(bps)
- ave rate to user(pps)
- sust rate from user(pps)
- sust rate to user(pps)
- ipv4 bad hdr
- ipv4 ttl exceeded
- ipv4 fragments sent
- ipv4 could not fragment
- ipv4 bad length trim
- ipv4 input mcast drop
- ipv4 input bcast drop
- input pkts dropped (0 mbr)
- output pkts dropped (0 mbr)
- ip source violations
- ipv4 output no-flow drop
- ipv6 bad hdr
- ipv6 bad length trim

- ipv4 input mcast drop
- ipv4 input bcast drop
- input pkts dropped (0 mbr)
- output pkts dropped (0 mbr)
- ip source violations
- ipv4 output no-flow drop
- ipv6 bad hdr
- ipv6 bad length trim
- ipv4 icmp packets dropped
- APN AMBR Input Pkts Drop
- APN AMBR Output Pkts Drop
- APN AMBR Input Bytes Drop
- APN AMBR Output Bytes Drop
- Total subscribers matching specified criteria

show subscribers user-plane-only seid *seid* pdr all

Executing this show command displays the following output:

- Source Interface
  - Type
- Destination Interface
  - Type
- vv
- PRD-ID
- Linked FAR-ID
- Linked URR-ID
- Linked QER-ID
- Total subscribers matching specified criteria

show user-plane-service [all | name *name*]

Executing this show command displays the following output:

- Service name

- Service-Id
- Context
- Status
- PGW Ingress GTPU Service
- SGW Ingress GTPU Service
- SGW Egress GTPU Service
- Control Plane Tunnel GTPU Service
- Sx Service

show user-plane-service statistics all

Executing this show command displays the following output:

- VPN Name
- Subscribers Total
    - PDNs Total
        - Active
        - Setup
        - Released
        - Rejected
- PDNs By PDN-Type
    - IPv4 PDNs
        - Active
        - Setup
        - Released
    - IPv6 PDNs
        - Active
        - Setup
        - Released
    - IPv4v6 PDNs
        - Active
        - Setup

- Released

- PDNs By interface-Type
  - Sxa interface-type PDNs
    - Active
    - Released
  - Sxb interface-type PDNs
    - Active
    - Setup
    - Released

- PDNs Rejected By Reason
  - No Resource
  - Missing or unknown APN
  - Addr not alloc
  - Addr not present
  - No memory available
  - System Failure
  - PDR install failed

- PDNs Released By Reason
  - Network initiated release
  - Admin disconnect

- Total Data Statistics
  - Uplink
    - Total Pkts
    - Total Bytes
    - Total Dropped Pkts
    - Total Dropped Bytes
  - Downlink
    - Total Pkts
    - Total Bytes
    - Total Dropped Pkts

- Total Dropped Bytes

- Data Statistics Per PDN-Type
  - IPv4 PDNs
    - Uplink
      - Total Pkts
      - Total Bytes
    - Downlink
      - Total Pkts
      - Total Bytes
  - IPv6 PDN Data Statistics
    - Uplink
      - Total Pkts
      - Total Bytes
    - Downlink
      - Total Pkts
      - Total Bytes
  - IPv4v6 PDN Data Statistics
    - Uplink
      - Total Pkts v4
      - Total Bytes v4
      - Total Pkts v6
      - Total Bytes v6
    - Downlink
      - Total Pkts v4
      - Total Bytes v4
      - Total Pkts v6
      - Total Bytes v6

- Flow Statistics
  - Max Flow Reached

- Pkts Dropped - system Limit (L4)
- Ip Flow Statistics
  - Total Flows v4
    - Uplink
    - Total Pkts v4
    - Total Bytes v4
    - Total Error Pkts v4
    - Total Error Bytes v4
  - Active Flows v4
    - Downlink
    - Total Pkts v4
    - Total Bytes v4
    - Total Error Pkts v4
    - Total Error Bytes v4
  - Total Flows v6
    - Uplink
    - Total Pkts v6
    - Total Bytes v6
    - Total Error Pkts v6
    - Total Error Bytes v6
  - Active Flows v6
    - Downlink
    - Total Pkts v6
    - Total Bytes v6
    - Total Error Pkts v6
    - Total Error Bytes v6

- Udp Flow Statistics
  - Total Udp Flows
    - Uplink
      - Total Udp Pkts

- Total Udp Bytes
- Total Udp Error Pkts
- Total Udp Error Bytes

- Downlink
  - Total Udp Pkts
  - Total Udp Bytes
  - Total Udp Error Pkts
  - Total Udp Error Bytes

- TCP Flow Statistics
  - Total TCP Flows
    - Uplink
      - Total TCP Pkts
      - Total TCP Bytes
      - Total TCP Error Pkts
      - Total TCP Error Bytes

    - Downlink
      - Total TCP Pkts
      - Total TCP Bytes
      - Total TCP Error Pkts
      - Total TCP Error Bytes

show user-plane-service statistics charging action

This command displays charging action statistics for all or specified charging actions that are configured in the Active Charging Service (ACS). A charging action represents actions to be taken when a configured rule is matched. Actions range from generating accounting records to dropping the IP packet, and so on. The charging action also determines the metering principle—Whether to count retransmitted packets, and which protocol field to use for billing (L3/L4/L7, and so on).

Syntax

show user-plane-service statistics charging-action
{ all [debug-info | verbose] | name *charging_action_name* **[debug-info | verbose] } [| { grep** *grep_options* **| more }]**

Notes:

- **all**: Displays information for all charging actions configured in ACS.
- **name** *charging_action_name*: Displays information for an existing charging action specified as an alphanumeric string from 1 through 63 characters.

This show CLI command doesn't support the following statistics with the value *0* that is displayed for each of its counter value.

```
PP Flows Readdressed:0
Bytes Charged Yet Packet Dropped:0
Predef-Rules Deactivated:0
Outer IP header dscp marked Pkts:0

Tethering Blocking Statistics:
   TTL Modified downlink packets:0

Throttle-Suppress Stats:
   Uplink Bytes:0    Downlink Bytes:0

XHeader Information:
IP Frags consumed by XHeader:0 IP Frags consumed by XHeader:0

Strip URL:
   Successful Token stripped:0
   Total strip URL failure:0
   Failure - Missing config:0
   Failure - Existing flow bid:0
   Failure - Token matching failed:0
   Failure - Empty packet:0
   Failure - Req end not found:0
   Failure - Subset of big token:0

URL-Readdressing:
   Requests URL-Readdressed:0
   Total Charging action hit - Req. Readdr.:0
   Proxy Disable Success:0
   Flows connected to URL Server:0


URL-Readdressing Error Conditions:
   Total connect failed to URL Server:0
   URL Readdress- pipelined case:0
   URL Readdress- Socket Mig. Failed:0
   Proxy Disable Failed:0

CAE-Readdressing:
   Requests CAE-Readdressed:0
   Responses CAE-Readdressed:0
```

```
  Requests having MVG xheader inserted:0
  Total CAE-Readdressed Uplink Bytes:0
  Total CAE-Readdressed Uplink Packets:0
  Total CAE-Readdressed Downlink Bytes:0
  Total CAE-Readdressed Downlink Packets:0
  Total Charging action hit - Req. Readdr.:0
  Total Charging action hit - Resp. Readdr:0
  Proxy Disable Success:0
  Flows connected to CAE:0

CAE Readdressing Error Conditions:
  Total connect failed to CAE:0
  Req. Readdr. - pipelined case:0
  Skipped Resp. Readdr. - pipelined req:0
  Req. Readdr. - Socket Mig. failed:0
  Skipped Resp. Readdr. - partial resp hdr:0
  Resp. Readdr. - Socket Mig. failed:0
  Total CAE load balancer failed:0
  Total MVG xheader insertion failed:0
  Proxy Disable Failed:0

  Rulebase Changed by flow action:0
  Terminate Session:0
  P2P random dropped packets:0
```

show user-plane-service statistics group-of-ruledefs

This command displays statistics for all groups or a specified group of **ruledef**s configured in the active charging service. The **group-of-ruledefs** is a collection of rule definitions that can be used in access policy creation.

Syntax

show user-plane-service statistics group-of-ruledefs { all | name *group_of_ruledefs_name* **} [| { grep** *grep_options* **| more }]**

Notes:

- **all**: Displays information for all **groups of ruledefs** configured in ACS.
- **name** *group_of_ruledefs_name*: Displays detailed information for an existing **group of ruledefs** specified as an alphanumeric string from 1 through 63 characters.
- **{ grep** *grep_options* **| more } Pipes**: Sends the output of this command to the specified command.
- The following clear CLI command is available for use:

  clear user-plane-service statistics group-of-ruledefs { all | name *group_of_ruledefs_name* **}**

show user-plane-service statistics ruledef

This command displays statistics for all or specified **ruledef** that is configured in an active charging service. The **ruledef** represents a set of matching conditions across multiple L3 - L7 protocol that is based on protocol fields and state information. You can use each **ruledef** across multiple rule bases within the active charging service.

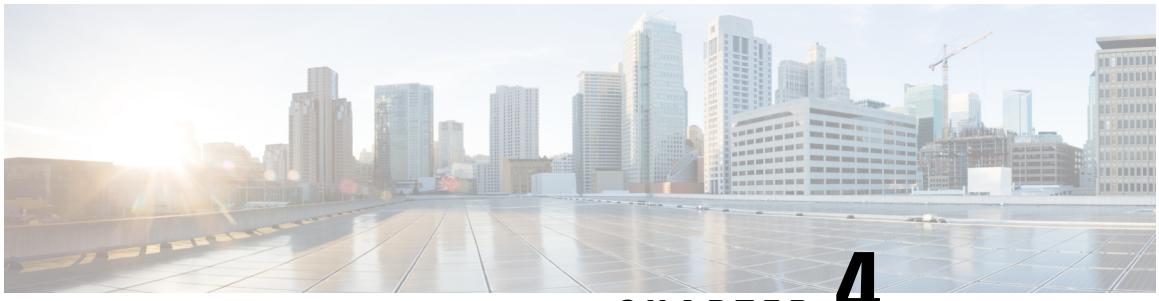
Syntax

show user-plane-service statistics ruledef { all { charging | firewall [wide] | post-processing } | name *ruledef_name* **[wide] } [| { grep** *grep_options* **| more }]**

Notes:

- **all**: Displays statistics for **all ruledef**s of the specified type that is configured in the ACS.
- **charging**: Displays statistics for all **charging ruledef**s configured in the ACS.
- **firewall**: Displays statistics for all **firewall ruledef**s configured in the service.
- **post processing**: Displays statistics for all **post processing ruledef**s configured in the ACS.
- **name** *ruledef_name*: Displays statistics for an existing **ruledef** specified as an alphanumeric string from 1 through 63 characters.
- **wide**: Displays all available information in a single wide line.
- The following clear CLI command is available for use:

  clear user-plane-service statistics ruledef { all | charging | firewall | name *group_of_ruledefs_name* **}**



CHAPTER 4

1:1 User Plane Redundancy for 4G CUPS

- Revision History, on page 69
- Feature Description, on page 69
- How it Works, on page 69
- Configuring 1:1 User Plane Redundancy for 4G CUPS, on page 79
- Monitoring and Troubleshooting, on page 84

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

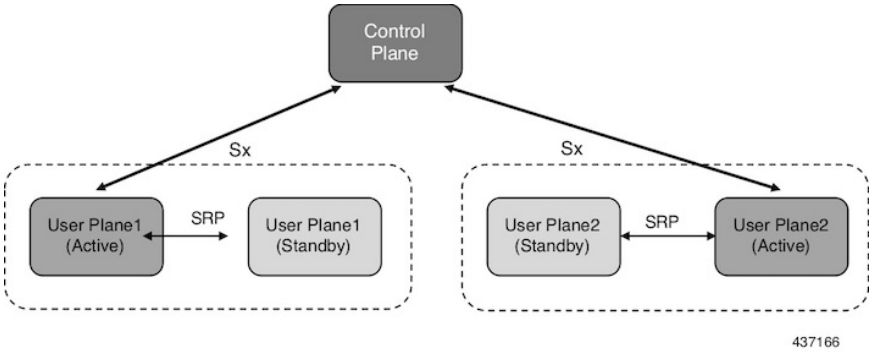
The 1:1 User Plane Redundancy for 4G CUPS feature supports the detection of a failed User Plane (UP) and handles seamlessly the functions of the failed UP. Each of the Active UPs has a dedicated Standby UP. The 1:1 UP redundancy architecture is based on the UP to UP Interchassis Session Recovery (ICSR) connection.

How it Works

This section briefly describes how 1:1 User Plane Redundancy for 4G CUPS feature works.

The 4G CUPS deployment leverages the ICSR framework infrastructure for checkpointing and switchover of the UP node as shown in the following figure. The Active UP communicates to its dedicated Standby UP via the Service Redundancy Protocol (SRP) link that is provisioned between the UPs.

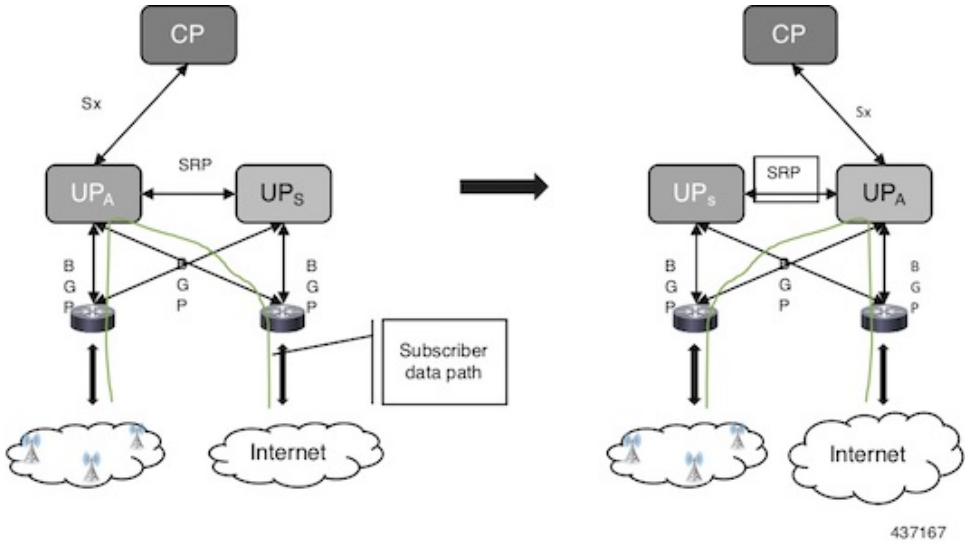
Figure 1: UP 1:1 Redundancy Using SRP



The Control Plane (CP) node does not have the Standby UP information that is available in the UP group configuration. Therefore, the CP is not aware of the UP redundancy configuration and the switchover event among the UPs.

The Active UP communicates to the CP via the Sx interface address configured in the UP. The Standby UP takes over the same Sx interface address when it transitions to the Active during the switchover event. This implies that the Sx interface is SRP activated and is in line with the existing configuration method, therefore UP switchover is transparent to the CP.

Figure 2: UP 1:1 Redundancy Switchover



To make redundancy fully compliant, it addresses the following dependencies on the SRP-based ICSR in the CUPS environment.

- Synchronization of PFD Configuration
- Sx Association Checkpoint
- Sx Link Monitoring

Besides the dependencies listed, the UP implements data collection and checkpoint procedures specific to the UP node. For example, checkpointing for IP-pool chunks. The UP integrates these procedures into the existing ICSR checkpointing framework.

Figure 3: CP-CP ICSR with 1:1 UP Redundancy, before CP Switchover

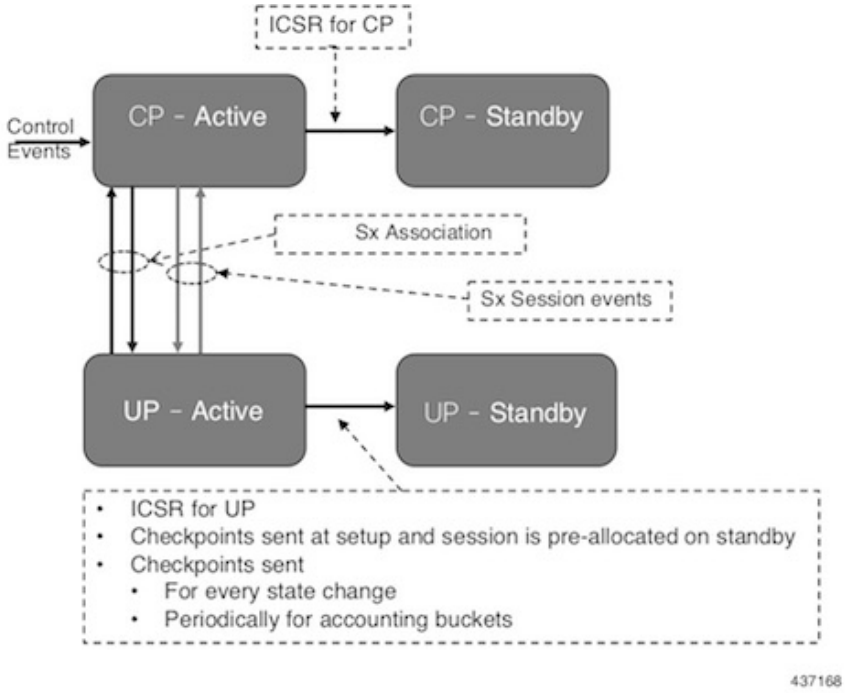
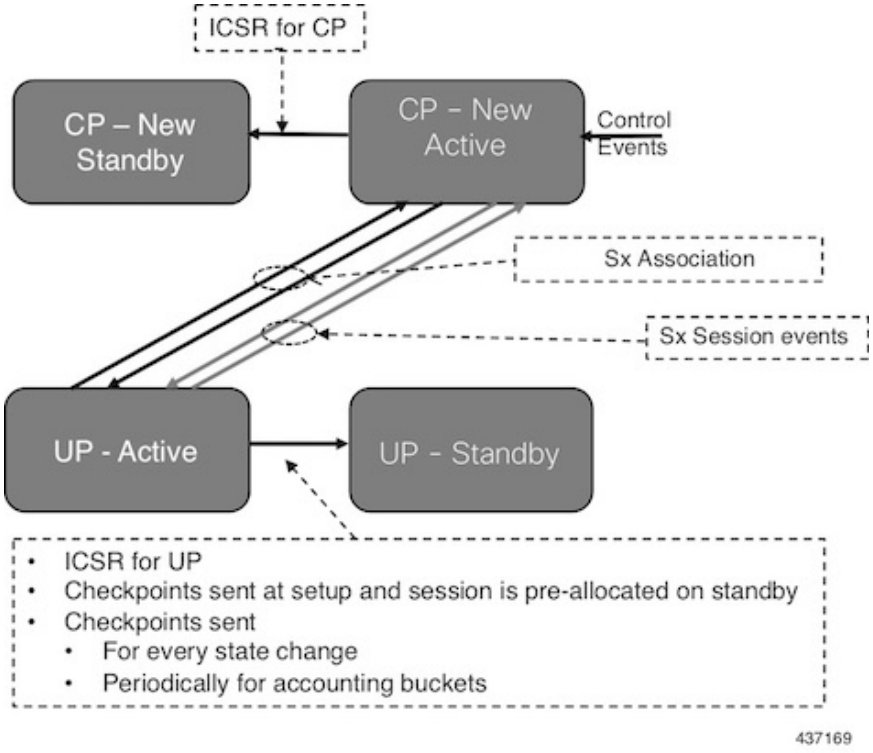


Figure 4: CP-CP ICSR with 1:1 UP Redundancy, After CP Switchover

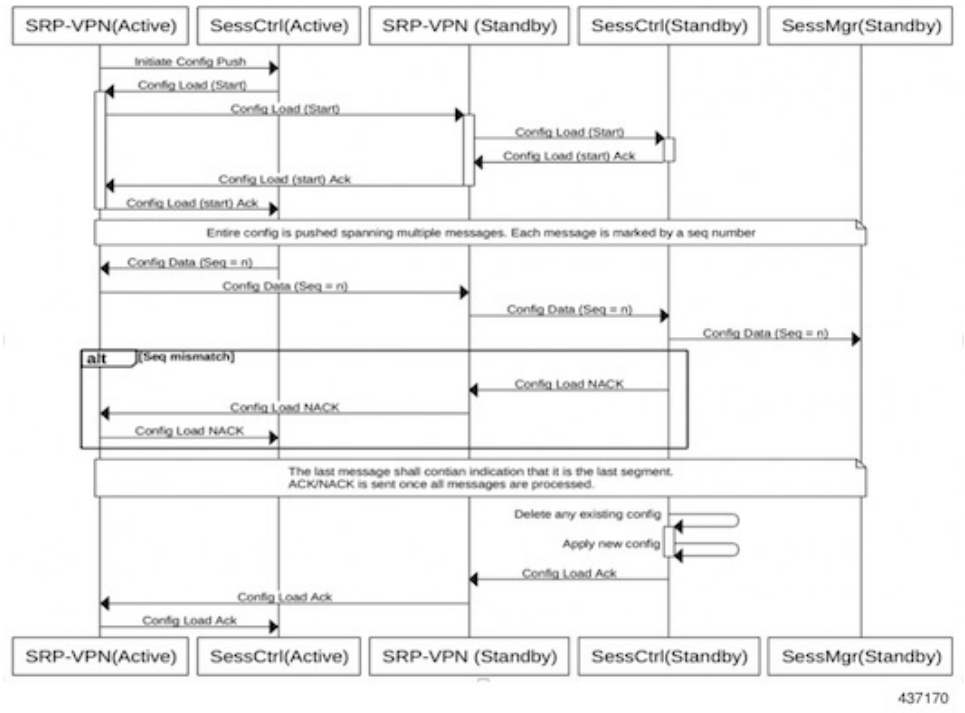


Synchronization of PFD Configuration

The CP node pushes the UP configuration via the Packet Flow Description (PFD) messages. The CP sends the PFD configuration from the Active UP to the Standby UP because the Sx IP address of the UP is SRP-activated over the Active UP and Standy UP.

The SRP VPN Manager provides the transport between UPs and the Session Controller in the Active UP anchors the configuration push. The following illustration lists the sequence of events.

Figure 5: Synchronizing PFD Configuration



BFD Monitor Between Active UP and Standby UP

The BFD monitors the SRP link between the Active UP and Standby UP for a fast failure detection and switchover. When the Standby UP detects a BFD failure in this link, it takes over as the Active UP.

The BFD link can be single-hop or multi-hop.

Note The recommendation is that the SRP bind interface must be an Ethernet interface that attaches to the card service Port. In a loopback address, the recommendation is to ensure that the BFD control packets traverse only though one service port. If it is the ECMP, ensure that the route convergence time does not exceed the BFD timeout.

To configure the BFD monitor, between the Active UP and Standby UP, see "Configuring BFD Monitoring Between Active UP and Standby UP."

Sample Configuration for Multihop BFD Monitoring

Primary UP:

```
config
  context srp
    bfd-protocol
      bfd multihop-peer 209.165.200.225 interval 50 min_rx 50 multiplier 20
    #exit
    service-redundancy-protocol
      monitor bfd context srp 209.165.200.225 chassis-to-chassis
      peer-ip-address 209.165.200.225
      bind address 209.165.200.227
```

```
    #exit
    interface srp
      ip address 209.165.200.227 255.255.255.224
    #exit
    ip route static multihop bfd bfd1 209.165.200.227 209.165.200.225
    ip route 192.168.210.0 255.255.255.224 209.165.200.228 srp
  #exit
end
```

Backup UP:

```
config
  context srp
    bfd-protocol
      bfd multihop-peer 209.165.200.227 interval 50 min_rx 50 multiplier 20
    #exit
    service-redundancy-protocol
      monitor bfd context srp 209.165.200.227 chassis-to-chassis
      peer-ip-address 209.165.200.227
      bind address 209.165.200.225
    #exit
    interface srp
      ip address 209.165.200.225 255.255.255.224
    #exit
    ip route static multihop bfd bfd1 209.165.200.225 209.165.200.227
    ip route 192.168.209.0 255.255.255.224 209.165.200.226 srp
  #exit
End
```

Router between Primary UP and backup UP:

```
config
  context one
    interface one
      ip address 209.165.200.228 255.255.255.224
    #exit
    interface two
      ip address 209.165.200.226 255.255.255.224
    #exit
  #exit
end
```

Sample Configuration for Single Hop BFD Monitoring

Primary UP:

```
config
  context srp
    bfd-protocol
    #exit
    service-redundancy-protocol
      monitor bfd context srp 255.255.255.230 chassis-to-chassis
      peer-ip-address 255.255.255.230
      bind address 209.165.200.227
    #exit
    interface srp
      ip address 209.165.200.227 255.255.255.224
      bfd interval 50 min_rx 50 multiplier 10
    #exit
    ip route static bfd srp 255.255.255.230
  #exit
end
```

Backup UP:

```
config
  context srp
    bfd-protocol
    #exit
    service-redundancy-protocol
      monitor bfd context srp 209.165.200.227 chassis-to-chassis
      peer-ip-address 209.165.200.227
      bind address 255.255.255.230
    #exit
    interface srp
      ip address 255.255.255.230 255.255.255.224
      bfd interval 50 min_rx 50 multiplier 10
    #exit
    ip route static bfd srp 209.165.200.227
  #exit
end
```

VPP Monitor

The SRP VPP monitor initiates a switchover to Standby UP when the VPP subsystem fails.

Note The VPP monitor is available only on the VPC-SI instance UP. It is not available in the hybrid CUPS ASR 5500 UP because the card level redundancy handles the VPP failure on the ASR 5500. If VPP causes multiple card failures, then SRP card monitor must be used.

To configure the VPP monitor, see "Configuring VPP Monitor on Active UP and Standby UP."

Sx Association Checkpoint

Whenever an Active UP initiates a Sx association to the configured CP node, the Standby UP checkpoints this data. This maintains the association information even after the UP switchover.

The Sx heartbeat messages sends and the Active UP must responds even after back-to-back UP switchovers.

Sx Monitor

It is critical to monitor the Sx interface between the UP and CP. Enabling the Sx heartbeat functionality is essential because it helps detect a monitor failure.

Note Sx monitoring is available only in the UP.

The Sx interface on the Active UP detects failure and informs the SRP VPN Manager to trigger the UP switchover event such that the Standby UP takes over.

It is important to ensure that the CP Sx heartbeat timeout is higher than the UP Sx heartbeat timeout plus UP ICSR switchover time. This is to ensure that the CP does not detect the Sx path failure during a UP switchover because of the UP Sx monitor failure.

Preventing Control Plane Heartbeat Time Out

There is a minor possibility that the CP heartbeat times out during the UP ICSR switchover. Follow these steps to mitigate it:

1. Remove the Sx heartbeats from the CP toward the UPs.

2. If the former is not possible, then ensure that the Sx heartbeats from the CP toward the UP have multiple retry timeout. Also ensure that the number of retries is greater than the UP Sx heartbeat timeout plus UP ICSR switchover time.

   For example:

   A = CP heartbeat interval (*sx-protocol heartbeat interval*)

   B = CP heartbeat max retransmissions (*sx-protocol heartbeat max-retransmissions*)

   C = CP heartbeat retransmission timeout (*sx-protocol heartbeat retransmission-timeout*)

   D = UP heartbeat interval (*sx-protocol heartbeat interval*)

   E = UP heartbeat max retransmissions (*sx-protocol heartbeat max-retransmissions*)

   F = UP heartbeat retransmission timeout (*sx-protocol heartbeat max-retransmissions*)

   G = Switchover time (including BGP route convergence time)

   Therefore, the formula for successful Sx monitor failure switchover is:

   B * C > D + (E * F) + G

   Example Values:

   CP:

   A:

   sx-protocol heartbeat interval 60

   B:

   sx-protocol heartbeat max-retransmissions 10

   C:

   sx-protocol heartbeat retransmission-timeout 10

   UP:

   D:

   sx-protocol heartbeat interval 30

   E:

   sx-protocol heartbeat max-retransmissions 3

   F:

   sx-protocol heartbeat retransmission-timeout 3

   BGP:

   G: Example route converge time = 30 sec

   Therefore, B * C > D + (E * F) + G

   => 10 * 10 > 30 + (3 * 3) + 30

   => 100 > 69

   A maximum value of B is 15 and max value of C is 20. Therefore, configure the Sx monitor failure detection and UP switchover (D + (E * F) + G) to withstand a maximum delay of 15 * 20 = 300 sec, that is, 5 min.

To minimize the BGP route convergence time (G), run the BGP with BFD fail-over.

To configure the Sx monitor, see "Configuring Sx Monitoring on the Active UP and Standby UP."

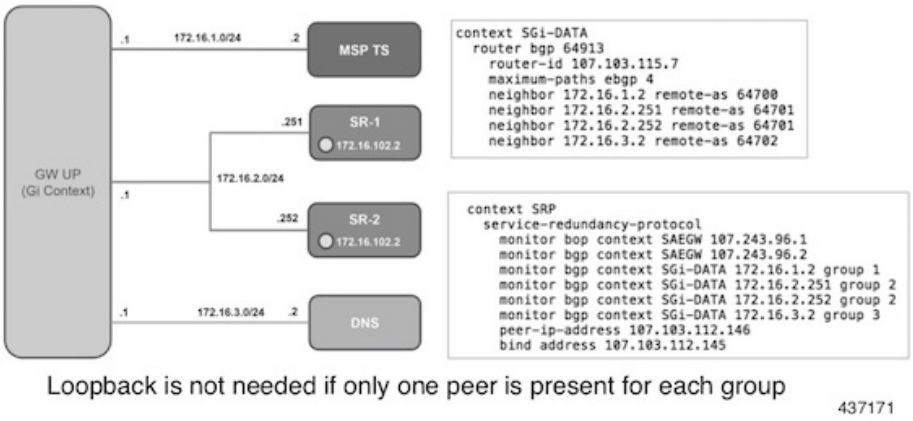
The Standby UP itself has no independent connectivity to the CP. The Active UP Sx context is replicated to the Standby UP such that it is ready to takeover during SRP switchover. This implies that when the Active UP has switched over to Standby because of Sx monitor failure, the new Standby has no way of knowing if the UP to CP link is working. To prevent a switchback of the new Standby to Active state again due to Sx monitor failure in new Active, use the **disallow-switchover-on-peer-monitor-fail** keyword in the new **monitor sx** CLI command.

After a chassis becomes Standby due to Sx monitoring failure, the Sx failure status is not reset even if Sx up checkpoint is received from the new Active UP. This is to prevent the new Active to cause an unplanned switchback again due to Sx monitor failure when the previous cause of switchover itself was Sx monitor failure. This prevents back-to-back ping-pong type of switchovers when CP is down. The Sx monitor failure status must be manually reset when the operator is convinced that the network connectivity is normal. To reset, use the new **srp reset-sx-fail** CLI command (see "Resetting Sx Monitor Failure") in the Standby chassis.

BGP Monitor

Configure BGP peer monitor and peer group monitors for the next-hop routers from UP (both Gi and Gn side) as shown in the following figure. This is the existing ICSR configuration. BGP may run with BFD assist to detect fast BGP peer failure.

Figure 6: BGP Peer Groups and Routing



To configure BGP monitoring and flag BPG monitoring failure, see Flagging BGP Monitoring Failure, on page 80.

UP Session Checkpoints

The Active chassis sends a collection of UP data as checkpoints to the peer Standby chassis in the following scenarios:

- New call setup
- For every state change in the call
- Periodically for accounting buckets

On receiving these checkpoints, the Standby chassis acts on the data and updates the necessary information either at the call level or node or instance level.

VPN IP Pool Checkpoints

Along with the PFD configuration message, the CP sends the IP pool allocation to each UP. The VPN manager receives this message in the UP and checkpoints the same information to the Standby UP when the SRP is configured.

The IP pool information is also sent during the SRP VPNMGR restart and during the SRP link down and up scenarios.

Validation of the presence of IP pool information in the Standby is vital before switchover. If the IP pool information is not present, then route advertisement is not possible. Therefore, traffic does not reach the UP.

External Audit and PFD Configuration Audit Interaction

The Active UP performs external audit and PFD configuration audit interaction. The Session Manager gets a start and complete notification of the PFD configuration audit. The Session Manager does not start the external audit if the PFD configuration audit is in progress. If the PFD configuration audit start notification arrives when the external audit is already underway, then the Session Manager raises a flag such that the external audit restarts when it completes. Restarting the external audit is necessary because it does not achieve its purpose if it occurs when the PFD audit is already underway.

Zero Accounting Loss for User Plane

Zero accounting loss feature is implemented on User Plane (UP) so that accounting-data/billing loss is reduced from 18 seconds, which is the default checkpoint time from Active UP to Standby UP, or for the configured accounting checkpoint time.

This change in UP is to support the Gz, Gy, VoGx, and RADIUS URRs. Only planned switchover is supported for zero accounting loss/URR data counters loss. This feature does not impact the current ICSR framework or the way checkpointing is done and recovered.

The Sx usage report is blocked during the “pending active state” till the chassis becomes Active.

Early PDU Recovery for UP Session Recovery

Early PDU Recovery feature overcomes the earlier limitation of Session Recovery feature wherein it did not prioritize the CRRs that were selected for recovery. All the CRRs were fetched from the AAAMgr and then the calls were recovered sequentially. The time taken to fetch all the CRRs was a major factor in the perceived delay during session recovery. When a failure occurred, the delay was sometimes very long if there were a lot of sessions in a Session Manager. Also, since the calls were recovered in no particular order, the idle sessions were sometimes recovered before active sessions.

Note The Early PDU Recovery feature can recover a maximum of 5 percent sessions.

Session Prioritization during Recovery

Prior to this release, the Session Recovery function did not prioritize the sessions selected for recovery, and loops through all the calls in the call recovery list and are recovered sequentially when the session recovery is triggered.

As part of Session Prioritisation during Recovery, a separate skiplist is maintained only for priority calls so that these records can be sent from AAAMgr immediately without going through the loop, thus leading to quicker recovery of the priority calls and reducing the data outage time.

There are two types of sessions at User Plane, prioritized sessions and normal sessions. Session is considered to be prioritized session based on message priority flag received from Control Plane and it is recovered first followed by normal calls.

These prioritized sessions also take priority in case of early PDU handling. The early PDU of normal calls will only initiate recovery when all prioritized sessions have been recovered.

In case of critical flush (GR), checkpoints for prioritized sessions are sent first followed by the normal calls. The data of all the calls (both normal and prioritized) are allowed during switchover.

Note The Control Plane is responsible to set the priority flags for all the calls. The User Plane uses the priority call details received from the Control Plane for the Session Prioritisation feature.

Configuring 1:1 User Plane Redundancy for 4G CUPS

The following sections provide information about the CLI commands available to enable or disable the feature.

Configuring BFD Monitoring Between Active UP and Standby UP

Use the following commands to configure Bidirectional Forwarding Detection (BFD) monitoring on the Active UP and Standby UP. This command is configured in the SRP Configuration Mode.

```
configure
   context context_name
     service-redundancy-protocol
        [ no ] monitor bfd context context_name { ipv4_address | ipv6_address }
{ chassis-to-chassis | chassis-to-router }
        exit
```

NOTES:

- **no**: Disables BFD monitoring on the Active and Standby UP.
- **context** *context_name* : Specifies the context that is used. It refers to the context where the BFD peer is configured (SRP context).

  context_name must be an existing context expressed as an alphanumeric string of 1 through 79 characters.
- *ipv4 _address* | *ipv6_address*: Defines the IP address of the BFD neighbor to be monitored, entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

  It refers to the IP address of the configured BFD (ICSR) peer.
- **chassis-to-chassis** | **chassis-to-router**:

  chassis-to-chassis: BFD runs between primary and backup chassis on non-SRP links.

  chassis-to-router: BFD runs between chassis and router.

Caution Do not use the **chassis-to-router** keyword for BFD monitoring on the SRP link between the Active UP and the Standby UP.

- This command is disabled by default.

Flagging BGP Monitoring Failure

Use the following commands to flag BGP monitor failure on a single BGP peer (User Plane) failure. This command is configured in the SRP Configuration Mode.

Note

- In this release, the **exclusive-failover** keyword is added to the existing **monitor bgp** CLI command as an alternate (new) algorithm to flag BGP monitoring failure.
- For more informnation about the **monitor bgp** CLI command in the "Service Redundancy Protocol Configuration Mode Commands" section command of the Command Reference Guide.
- Before adding the **exclusive-failover** keyword to the existing **monitor bgp** CLI command, implementing the **monitor bgp** command resulted in the following behavior:
  - BGP peer group was up if any BGP peer in that group was up.
  - Omitting a group configuration for a BGP monitor included that monitor in group 0.
  - BGP group 0 monitored in a context from an implicit group. Each context formed a separate BGP group 0 implicit monitor group.
  - BGP monitor was down if any BGP peer group was down.

```
configure
  context context_name
    service-redundancy-protocol
      [ no ] monitor bgp exclusive-failover
      end
```

NOTES:

- **no**: Disables flagging of BGP monitor failure on a single BGP peer failure.
- On implementing the new **exclusive-failover** keyword, the behavior is as follows:
  - BGP peer group is Up if any BGP peer in that group is Up.
  - Including a BGP peer in group 0 is same as making it non-group (omitting group).
  - BGP monitor is down if any BGP peer group or any non-group BGP peer is down.
  - Removing a BGP peer being monitored induces a BGP monitor failure.
- This command is disabled by default.

Configuring Sx Monitoring on the Active UP and Standby UP

Use the following commands to configure Sx monitoring on the Active UP and Standby UP. This command is configured in the SRP Configuration Mode.

```
configure
   context context_name
      service-redundancy-protocol
        [ no ] monitor sx [ { context context_name | bind-address {ipv4_address
 |ipv6_address } | { peer-address {ipv4_address | ipv6_address } } ]
         exit
```

NOTES:

- **no**: Disables Sx monitoring on the Active and Standby UP.
- **context***context_name* : Specifies the context of the Sx service.

  context_name must be an existing context expressed as an alphanumeric string of 1 through 79 characters.
- **bind-address {** *ipv4 _address* **|** *ipv6_address***}**: Defines the service IP address of the Sx service, entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

Note The IP address family of the **bind-address** and **peer-address** must be same.

- **peer-address {** *ipv4 _address* **|** *ipv6_address***}**: Defines the IP address of the Sx peer, entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- **disallow-switchover-on-peer-monitor-fail** :

  Prevents the switchback of the UP to Active state when the working status of the UP to CP link is unknown.
- It is possible to implement this CLI command multiple times for monitoring multiple Sx connections.
- The Sx monitor state goes down when any of the monitored Sx connections are down.
- This command is disabled by default.

Configuring SRP over IPSec on the Active UP and Standby UP

IPSec is a suite of protocols that interact with one another to provide secure private communications across IP networks. These protocols allow the system to establish and maintain secure tunnels with peer security gateways. IPSec provides confidentiality, data integrity, access control, and data source authentication to IP datagrams.

The CUPS architecture uses the IPSec protocol to encrypt the packets sent over the Interchassis Session Recovery (ICSR) connection between the active and standby UPs. This encryption is done by defining an access-list to match all traffic between Service Redundancy Protocol (SRP) peers and associating it with a crypto map. This crypto map is used to establish Security Association between IPSec peers residing in UPs.

Note For more information on IPSec, its features or functionality, and applicable CLI configurations, refer the StarOS *IPSec Reference*.

The following CLI command is a sample configuration to configure SRP over IPSec for UPs.

```
context srp
   ip access-list srp-acl
   permit tcp host 209.165.200.225 host 209.165.200.226
   #exit
   ipsec transform-set A-foo
   #exit
   ikev2-ikesa transform-set ikesa-foo
   #exit
   crypto map srp-cm ikev2-ipv4
   match address srp-acl
   authentication local pre-shared-key key local key
   authentication remote pre-shared-key key remote key
   ikev2-ikesa transform-set list ikesa-foo
   payload foo-sa0 match ipv4
   ipsec transform-set list A-foo
   #exit
   peer 209.165.200.227
   #exit
   service-redundancy-protocol
   checkpoint session duration non-ims-session 30
   checkpoint session duration ims-session 30
   route-modifier threshold 18
   delta-route-modifier 2
   audit periodicity 60
   priority 2
   monitor bgp context isp 209.165.200.228
   monitor sx context EPC2 bind-address bbbb:abcd::77 peer-address bbbb:abcd::10
   peer-ip-address 209.165.200.226
   bind address 209.165.200.225
   #exit
   interface ike-lb loopback
   ip address 209.165.200.228 255.255.255.224
   crypto-map srp-cm
   #exit
   interface srp-rtr
   ip address 209.165.200.229 255.255.255.224
   #exit
   interface srp-loopback loopback
   ip address 209.165.200.225 255.255.255.224
   #exit
   ip route 209.165.200.226 255.255.255.224 209.165.200.231 srp-rtr
   ip route 209.165.200.227 255.255.255.224 209.165.200.231 srp-rtr
   #exit
```

Note IKEv1 - Transport mode with Authentication Header (AH) protocol is not recommended. Encapsulating Security Payload (ESP) is recommended because ESP performs both Authentication and Encryption.

Configuring VPP Monitor on Active UP and Standby UP

Use the following commands to configure Vector Packet Processing (VPP) monitor to trigger UP switchover on the Active UP if VPP goes down. This command is configured in the SRP Configuration Mode.

```
configure
   context context_name
      service-redundancy-protocol
         monitor system vpp delay-period 0-300 seconds
         exit
```

```
no monitor system vpp
```

NOTES:

- **no**: Disables VPP monitoring on the Active and Standby UP.
- **vpp delay-period***0-300 seconds* : Specifies the delay period in seconds for a switchover, after a VPP failure.

  If the delay period is a value greater than zero, then the switchover is initiated after the specified delay period when VPP fails. The last VPP status notification within the delay period is the final trigger for switchover action. The default value is 0 seconds, which initiates an immediate switchover.

  The need for delay is to address the scenario wherein the VPP is temporarily down and the revival is in process. This implies that a switchover may not be necessary.
- This command is disabled by default.

Preventing User Plane Switchback

Use the following commands to prevent the switchback of the new Standby UP to Active state again due to Sx monitor failure in the new Active. This command is configured in the SRP Configuration Mode.

```
configure
   context context_name
      service-redundancy-protocol
         monitor sx disallow-switchover-on-peer-monitor-fail [ timeout
seconds ]
         exit
```

Use either of the following CLIs to allow switchback of the new Standby UP to Active state.

```
no monitor sx disallow-switchover-on-peer-monitor-fail
```

Or

```
monitor sx disallow-switchover-on-peer-monitor-fail timeout 0
```

NOTES:

- **no**: Disables prevention of switchover.
- **disallow-switchover-on-peer-monitor-fail [timeout** *seconds* **]** : Prevents the switchback of the UP to Active state when the working status of the UP to CP link is unknown.

timeout *seconds*: Timeout after which the switchback is allowed even if the Sx failure status is not reset in the Standby peer. The valid values range from 0 to 2073600 (24 days).

Note Assigning 0 seconds as the the timeout allows unplanned switchover.

If **timeout** keyword is not specified, the Active chassis waits indefinitely for the Sx failure status to be reset in the Standby peer.

- The default configuration is to allow unplanned switchover due to Sx monitor failure in all conditions.

Note Manual planned switchover is allowed irrespective of whether this CLI is configured or not.

Preventing Dual Active Error Scenarios

Use the following CLI configuration in CP to prevent dual Active error scenarios for UP 1:1 redundancy.

```
configure
    user-plane-group group_name
        sx-reassociation disabled
        end
```

NOTE:

- **sx-reassociation disabled**: Disables UP Sx reassociation when the association already exists with the CP.

Resetting Sx Monitor Failure

Use the following command only on the Standby chassis to reset the Service Redundancy Protocol (SRP) Sx monitor failure information. This command is configured in the Exec Mode.

```
srp reset-sx-fail
```

Monitoring and Troubleshooting

This section provides information regarding the CLI command available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show srp monitor bfd

The output of this CLI command contains the following fields for the 4G CUPS 1:1 UP Redundancy feature:

- Type
- State
- GroupId
- IP Addr
- Port
- Context (VRF Name)
- Last Update

show srp monitor bgp

The output of this CLI command contains the following fields for the 4G CUPS 1:1 UP Redundancy feature:

- Type
- State
- GroupId
- IP Addr
- Port
- Context (VRF Name)
- Last Update

show srp monitor sx

The output of this CLI command contains the following fields for the 4G CUPS 1:1 UP Redundancy feature:

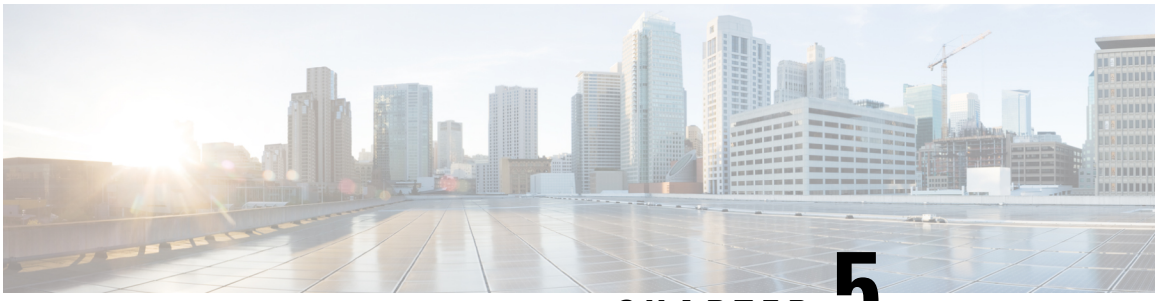
- Type
- State
- GroupId
- IP Addr
- Port
- Context (VRF Name)
- Last Update

show srp monitor vpp

The output of this CLI command contains the following fields for the 4G CUPS 1:1 UP Redundancy feature:

- Type

- State
- GroupId
- IP Addr
- Port
- Context (VRF Name)
- Last Update



CHAPTER 5

5G NSA for SAEGW in CUPS

- Feature Description, on page 87

Feature Description

Cisco 5G Non Standalone (NSA) solution leverages the existing LTE radio access and core network (EPC) as an anchor for mobility management and coverage. This solution enables operators using the Cisco EPC Packet Core to launch 5G services in shorter time and leverage existing infrastructure. Thus, NSA provides a seamless option to deploy 5G services with very less disruption in the network.

5G is the next generation of 3GPP technology, after 4G/LTE, defined for wireless mobile data communication. The 5G standards are introduced in 3GPP Release 15 to cater to the needs of 5G networks.

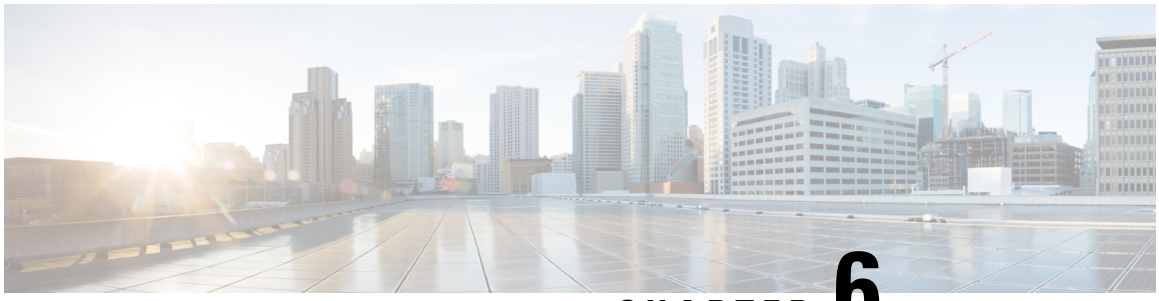
5G Non Standalone (NSA): The existing LTE radio access and core network (EPC) is leveraged to anchor the 5G NR using the Dual Connectivity feature. This solution enables operators to provide 5G services with shorter time and lesser cost.

Limitation

- In CUPS architecture, the SGW-C/PGW-C selecting SGW-U/PGW-U based on DCNR is not supported in this release.
- In this release, APNMBR rate-limit configuration is not supported. The APNMBR policer uses Auto-readjust internally.

  For more information on limitations, refer to the *5G NSA for SAEGW* chapter in the *5G Non Standalone Solution Guide*

For additional information about 5G NSA for SAEGW, refer the *5G NSA for SAEGW* chapter in the *5G Non Standalone Solution Guide*.



CHAPTER 6

Access Control Lists

- Revision History, on page 89
- Feature Description, on page 89
- Configuring Access Control Lists, on page 89
- Monitoring and Troubleshooting, on page 90

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The CUPS architecture supports Access Control Lists on the User-Plane. This feature allows the User-Plane to create and manage IP access privileges for a subscriber.

Configuring Access Control Lists

An existing configuration, which is part of the non-CUPS architecture is implemented for this feature. The **ip access-list** command – part of the Context Configuration mode is used to implement an access control list.

Note For CUPS, the same configuration is implemented on a User Plane's APN Configuration mode.

Use the following configuration to create and manage IP-based, user access privileges:

```
configure
  context context_name
```

```
ip access-list acl_name
   { deny | permit } [ log ] source_address source_wildcard
   no { deny | permit } [ log ] source_address source_wildcard
   end
```

NOTES:

- **no**: Removes the rule which exactly matches the options specified.
- **deny | permit**: Specifies the rule is either block (deny) or an allow (permit) filter.
  - **deny**: Indicates the rule, when matched, drops the corresponding packets.
  - **permit**: Indicates the rule, when matched, allows the corresponding packets.
- **log**: Indicates all packets which match the filter are to be logged. By default, packets are not logged.
  - *source_address*: The IP address(es) from which the packet originated. IP addresses must be entered in IPv4 dotted-decimal format.

    This option is used to filter all packets from a specific IP address or a group of IP addresses.

    When specifying a group of addresses, the initial address is configured using this option. The range can then be configured using the *source_wildcard* parameter.
  - *source_wildcard*: This option is used in conjunction with the *source_address* option to specify a group of addresses for which packets are to be filtered.

    The mask must be entered as a complement:
    - Zero-bits in this parameter mean that the corresponding bits configured for the *source_address* parameter must be identical.
    - One-bits in this parameter mean that the corresponding bits configured for the *source_address* parameter must be ignored.

Note The mask must contain a contiguous set of one-bits from the least significant bit (LSB). Therefore, allowed masks are 0, 1, 3, 7, 15, 31, 63, 127, and 255. For example, acceptable wildcards are 0.0.0.3, 0.0.0.255, and 0.0.15.255. A wildcard of 0.0.7.15 is not acceptable since the one-bits are not contiguous.

Monitoring and Troubleshooting

This section provides information regarding monitoring and troubleshooting the Access Control Lists feature.

Show Command(s) and/or Outputs

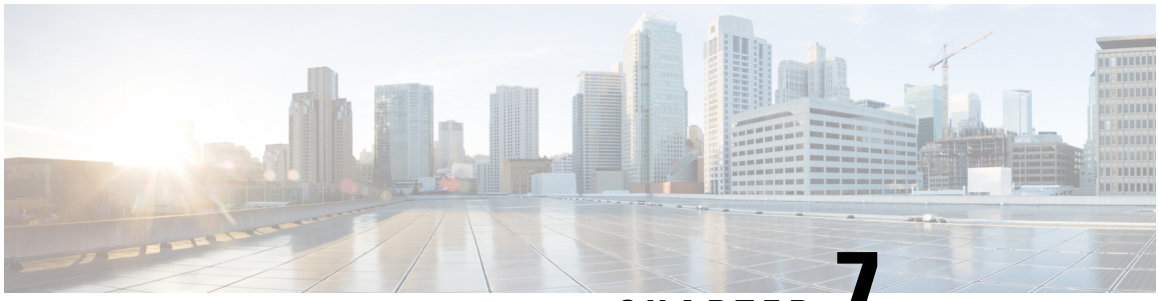
This section provides information regarding show commands and/or their outputs in support of this feature.

show sub user-plane-only full all

On executing the above command, the following fields are displayed for this feature:

- active input acl
- active output acl
- ipv4 input acl drop
- ipv4 output acl drop

show sub user-plane-only full all



CHAPTER 7

ADC Over Gx

- Feature Description, on page 93
- How It Works, on page 94
- Configuring ADC over Gx, on page 95
- Monitoring and Troubleshooting, on page 96

Feature Description

In compliance with 3GPP TS 29.244 V15.0.0, ADC over Gx feature supports the following functionalities in CUPS environment:

- Application START/STOP event reporting at the instance level, over the Sx Interface, as part of the session usage report request.
- Application START/STOP is sent for Group of Ruledef when a flow matches the Group of Ruledef.
- Supports AND logic for rulelines while matching ADC ruledefs.
- Supports new Information Elements (IEs) for Packet Forwarding Control Protocol (PFCP) messages that are used for ADC application detection notifications.

Important In this release, the ADC Over Gx feature is applicable for ADC L3/L4 rules.

Important For supplemental information about ADC Over Gx feature in non-CUPS environment, refer to:

- *ADC Support over Gx* section in the *Application Detection and Control Overview* chapter of the *ADC Administration Guide*.
- *Support ADC Rules over Gx Interface* section in the *Gx Interface Support* chapter of the *P-GW Administration Guide*.

How It Works

For ADC over Gx feature in the CUPS environment, support is added for the following:

- The Application ID/TDF Application Identifier is part of the PDI of PDR, either in Sx Establishment Request or Sx Session Modify Request.
- Handling of the ADC rule match on U-Plane.
- To generate a session usage report request when the Application START/STOP event occurs on U-Plane.
- New IEs as part of the Usage report request:
  - Application ID
  - Application Instance ID
  - Flow Information
- Monitor Protocol to decode the new IEs.
- To handle the usage report request that is received, and trigger the CCR-U to PCRF on C-Plane.

The functionality of ADC Over Gx feature consists of the following components, and each are described in this section.

ADC Rule Match

The ADC rule match is invoked after the traditional rule match. After the L3/L4 filters are being matched, the rule match engine checks for any ADC rules being configured on the bearer. If ADC rules are present, then the ADC rule match occurs.

If the bearer has ADC rule which does not have the L3/L4 filters, and it's a non-GBR bearer, then the ADC rule match is done across all the non-GBR bearers. The charging is done against the charging and action policy of the rule match.

For ADC dynamic rules, if the L3/L4 filter matches but the ADC rule match fails, then the rule is considered as not matched.

Session Usage Report Request Generation

Once the ADC rule matches on the U-Plane and an application has been detected, the U-Plane triggers the Application START notification over the Sx interface as a session usage report:

- With the measurement method set to Event
- Usage Report Trigger set to "Start of Traffic"
- The Application Detection Info, such as Application ID, Application Instance ID, and Application Flow Information along with the direction.

When the application teardown gracefully, the application gets timed out, or the rule match changes, the application STOP is triggered from U-Plane to C-Plane as a session usage report:

- With the measurement method set to "Event"

- Usage Report Trigger set to “Stop of Traffic”
- Application ID
- Application Instance ID

The application STOP is not triggered when:

- “mute” is enabled.
- The call is going down.
- The rule/PDR is deleted.
- The bearer/tunnel deletion occurs.

Handling Session Usage Report on C-Plane

After receiving the session usage report on C-Plane, it detects the event and CCR-U is triggered toward PCRF, along with the required attributes to be sent.

Limitations

Following are the known limitations of the ADC Over Gx feature:

- When the TDF Application Identifier on the U-Plane and the “**policy-control bypass TDF-ID-validation** CLI command are not present, the calls are dropped. And, the proper disconnect reason is not being shown.
- The configuration change for predefined ADC rules, such as "mute" to "unmute" and "unmute" to "mute" scenarios are not supported in this release.
- Mid-session update and/or modification of ADC rules—whether change in configuration or PDN update over RAR, is not supported.
- ADC is supported for L3/L4 rules on default bearer.

Licensing

ADC over Gx feature requires Application Detection Control License. Contact your Cisco account representative for detailed information on specific licensing requirements.

Configuring ADC over Gx

The CLI commands available for ADC Over Gx in non-CUPS environment can be used in CUPS environment.

Following are the sample configurations to:

- Enable the feature under Policy Control Configuration mode:

```
diameter encode-supported-features adc-rules
```

- Configure ADC predefined rule under ACS Rulebase Configuration mode:

```
action priority 55 dynamic-only adc ruledef qci5 charging-action charge-action-qci5
action priority 56 dynamic-only adc mute group-of-ruledefs qci5_gor charging-action
charge-action-qci5
```

Important Application START/STOP will not be sent to PCRF if the Application START/STOP event trigger is not registered while enabling the ADC Over Gx feature.

Important For additional information about the CLI commands, refer the *Command Line Interface Reference*.

Monitoring and Troubleshooting

This section describes the CLI commands available to monitor and/or troubleshoot the feature.

Monitor Protocol

When using the monitor protocol command, enable option 49 to see ADC related parameters in Sx messages.

Show Command(s) and/or Outputs

On C-Plane

show active-charging subscribers callid <callid> urr-info

The output of this show command has been modified to display the ADC URRs along with Volume and Duration related URRs.

On U-Plane

show subscribers user-plane-only full all

The output of this show command has been modified to display the “Number of associated ADC PDRs”.

show subscribers user-plane-only callid <callid> pdr full all

The output of this show command has been modified to display the following new fields:

- TDF App Id
- TDF Notifications
- Total ADC PDRs found

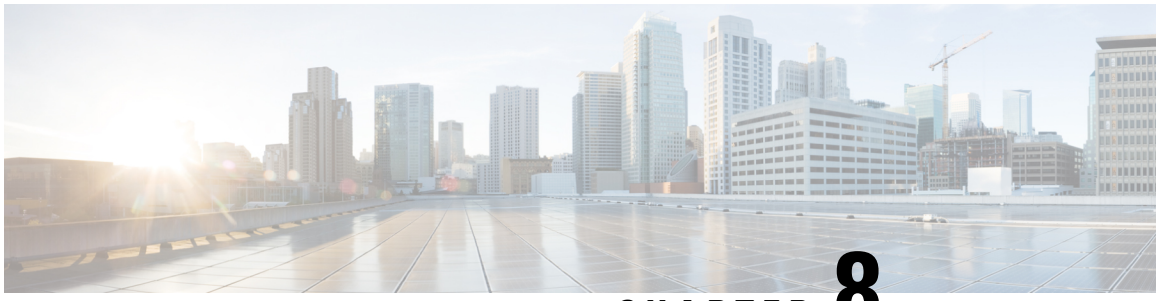
show subscribers user-plane-only callid <callid> urr full all

The output of this show command has been modified to display the ADC URRs along with Volume and Duration related URRs.

show user-plane-service rulebase name <rulebase_name>

The output of this show command has been enhanced in support of this feature. Two new Type characters are introduced to identify ADC rules and ADC rules with “mute”:

- RDA – Where A is for ADC rule
- GDAM – Where AM is for ADC rule with “mute”



CHAPTER 8

Addition of IP Pool in IP Group

- Revision History, on page 99
- Feature Description, on page 99
- How it Works, on page 100
- Monitoring and Troubleshooting, on page 100

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In the existing CUPS platform, when a new IP pool is added, only the User Planes (UPs) that register after the creation of the new pool can use these pools. If any existing UP requires to use the new pool, a UP reload or UP reassocation is performed.

The Addition of IP Pool in IP Group feature ensures that when a new IP pool is added, each existing UP is evaluated based on whether its APN configuration makes it eligible to get chunks from this new pool. If the UP is eligible, then chunks are allocated to the UP and it is used for future call allocation.

The eligibility of the UP is determined in the following scenarios:

- APN has a pool-group configured. A new pool is added under this pool-group.
- APN has no pool-name or pool-group configured . A new public pool is added.

Note Any changes implemented on the APN do not take affect until the UP is reassociated or reloaded.

How it Works

This section briefly describes how the Addition of IP Pool in IP Group feature works.

Adding New Pools in a CP-CP ICSR Environment

1. Add the new pool in the Standby Control Plane (CP).
2. Add the new pool in the Active CP.

   Chunks are allocated to the eligible UPs and the same are checkpointed to the Standby CP.
3. Verify whether **show { ip | ipv6 } pool-chunks pool-name** <*name*> command in both the CPs are synchronized.

Delete Pools in CP-CP ICSR Environment

1. Delete the pool in the Active CP.
2. Ensure that all the IPs are free from the deleted pool in the Standby CP, using the **show { ip | ipv6 } pools** command.
3. Delete the pool in the Standby CP.

Monitoring and Troubleshooting

This section provides information regarding the CLI command available in support of monitoring and troubleshooting the feature.

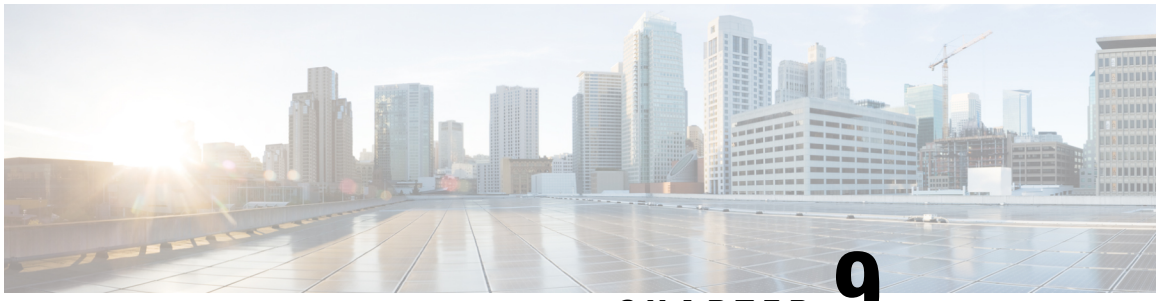
Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show ip user-plane verbose

The output of this CLI command displays the following fields in support of the Addition of IP Pool in IP Group feature in CUPS mode:

- Dynamic pool count
- apn-without-pool-name-v4
- apn-without-pool-name-v6
- Pool-groups
- Pool-Group-Names



CHAPTER 9

APN ACL Support

- Revision History, on page 101
- Feature Description, on page 101
- Troubleshooting, on page 102

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Currently in CUPS (pre 21.19.x release), the APN level ACL definitions are configured on UP.

With this feature, ACLs configured on CP are pushed to UP. This feature saves the cost and effort of configuring separate ACL definitions on all UP nodes.

Note

- Verify the APN ACLs in CP configuration before proceeding with the upgrade to the release.
- The configuration must have the same context names in both CP and UP. CP can have more contexts than UP. If the context names do not match, the respective ACLs are dropped at UP.
- It is recommended to not define APN ACLs in both CP and UP. However, if there is a requirement, the ACL names in both UP and CP must be different from each other to avoid any conflicts.
- To ensure backward compatibility, ACLs locally created in UP configuration gets preference.
- If an APN belongs to a specific user-plane-group, ACLs for the same APN are pushed to only those UPs, which are part of the same user-plane-group.
- A maximum of 64 contexts is allowed and a maximum of 16 ACLs per context.
- Multiple APNs can share an ACL in the same context.
- Changes to an ACL are applicable only for new sessions, but not for ongoing sessions.
- If a **deny any** rule is configured in IPv6 ACLs, the Router advertisement (RA) and Router Solicitation (RS) messages must be explicitly allowed in ACL.

Troubleshooting

This section describes how to troubleshoot this feature.

Note This feature is enabled by default.

Show commands

This section describes the show commands for this feature.

show user-plane-service ip-access-list name *access list name*

This command is used to display ACL rules on user plane.

show user-plane-service pdn-instance name *apn name*

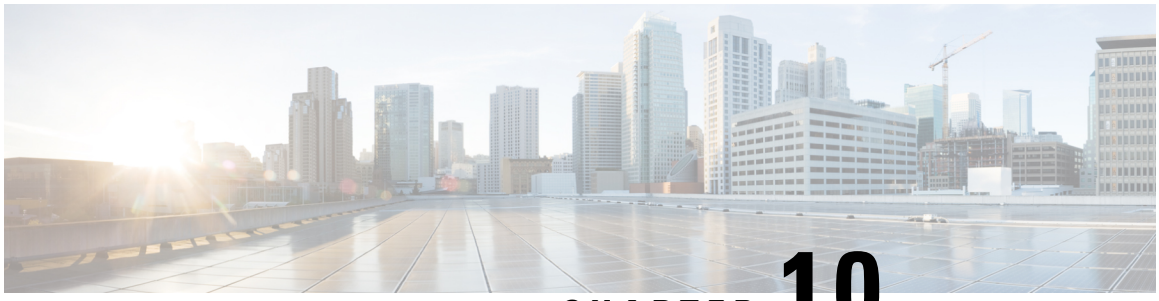
This command is used to display the access group for an apn on user plane.

show srp statistics

This command is used to display the sent, received, and discarded packet count for APN ACLs over SRP.

show demux-mgr statistics sxdemux all

This show command is used to display the number of PFD ACL_INFO packets sent from CP.



CHAPTER 10

APN AMBR Traffic Policing

- Revision History, on page 103
- Feature Description, on page 103
- Configuring the APN AMBR Traffic Policing Feature, on page 104
- Monitoring and Troubleshooting, on page 104

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The APN-AMBR is a subscription parameter stored per APN in the HSS. S-GW provides APN-AMBR during default bearer establishment procedure. APN-AMBR limits the aggregate bit rate that can be expected to be provided across all non-GBR bearers and across all PDN connections of the same APN. Each of those non-GBR bearers could potentially utilize the entire APN-AMBR, for example, when the other non-GBR bearers don't carry any traffic. The P-GW enforces the APN-AMBR in downlink and uplink direction.

As part of this CLI-controlled feature, the CLI parameters must be configured on Control Plane and propagated to User Plane through Sx interface.

Limitations

The following is the known limitation of APN-AMBR Traffic Policing feature:

- Configuring **token-replenishment-interval** and **violate-action shape** CLIs aren't supported.

Configuring the APN AMBR Traffic Policing Feature

This section describes how to configure the APN-AMBR Traffic Policing feature.

```
configure
         context context_name
           apn apn_name
       apn-ambr rate-limit direction { downlink | uplink } [ burst-size
{ auto-readjust duration { milliseconds msecs | seconds } | violate-action
 { drop | lower-ip-precedence | transmit }
                end
```

NOTES:

- **rate-limit direction { downlink | uplink }**: Specifies that the rate limit is to be applied to either the downlink (network to subscriber) traffic or the uplink (subscriber to network) traffic.
- **burst-size { auto-readjust duration milliseconds msecs | seconds }**: This parameter is used by policing algorithms to permit short bursts of traffic not to exceed the allowed data rates. It's the maximum size of the token bucket.
  - **auto-readjust duration***seconds*: The duration (in seconds) used in this burst size calculation: burst size = peak data rate/8 * auto-readjust duration.
    - Seconds must be an integer value from 1-30. Default is 1 second.
  - **milliseconds**: *msecs* must be an integer value from 100-900, in increments of 100 milliseconds. For example, 100, 200, or 300, and so on.
- **violate-action { drop | lower-ip-precedence | transmit }**: The action that the P-GW takes when the data rate of the bearer context exceeds the AMBR.
  - **drop**: Drops violating packets.
  - **lower-ip-precedence**: Sets the DSCP value to zero ("best effort") for violating packets.
  - **transmit**: Transmits violating packets. This is the default behavior of the feature.
- Prior to this feature, the default behavior was to drop the violating packets.

Monitoring and Troubleshooting

This section provides information about the commands available to monitor and/or troubleshoot the APN-AMBR Traffic Policing feature.

Show Commands and or Outputs

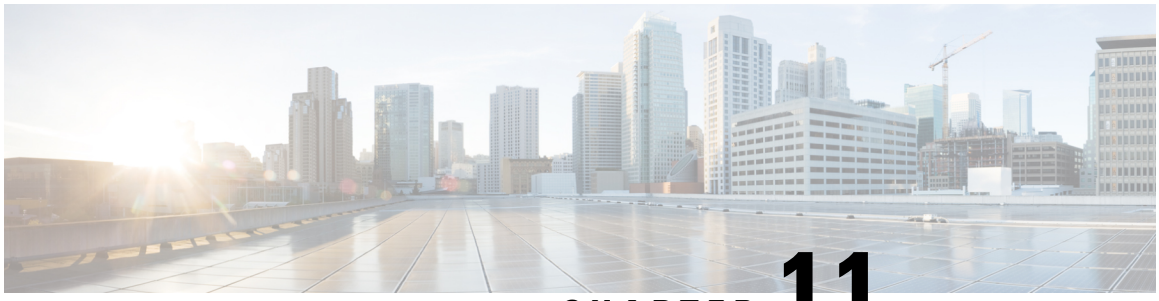
This section provides information about the show commands available for monitoring and/or troubleshooting the APN-AMBR Traffic Policing feature.

- **show user-plane-service pdn-instance name <apn_name>**: The following APN-AMBR information is available on User Plane after APN-AMBR CLI is configured on Control Plane and PFD Push to User Plane is completed:
  - APN-AMBR
    - Downlink Apn Ambr: Indicates if the rate limit is enabled or disabled for downlink traffic.
      - Burst Size: Indicates the burst size of the downlink traffic.
      - Auto Readjust: Indicates if the auto-readjust is enabled or disabled for downlink burst size.
      - Auto Readjust Duration: Indicates the duration used in downlink burst size calculation.
      - Burst Size(bytes): Indicates the burst size in bytes.
      - Violate Action: Indicates the action that the P-GW takes when the data rate of the bearer context exceeds the AMBR for downlink traffic.
    - Uplink Apn Ambr: Indicates if the rate limit is enabled or disabled for uplink traffic.
      - Burst Size: Indicates the burst size of the uplink traffic.
      - Auto Readjust: Indicates if the auto-readjust is enabled or disabled for uplink burst size.
      - Auto Readjust Duration: Indicates the duration used in uplink burst size calculation.
      - Burst Size(bytes): Indicates the burst size in bytes.
      - Violate Action: Indicates the action that the P-GW takes when the data rate of the bearer context exceeds the AMBR for uplink traffic.
    - Token Replenishment Interval: Indicates the token replenishment interval duration.

- **show sub user-plane-only full all**:

  Use this show command in User Plane to see the count of packets that are dropped, and IP precedence lowered due to APN-AMBR policer. The following fields are introduced in support of this feature:
  - APN AMBR Uplink Pkts Drop: Indicates the number of APN-AMBR packets that are dropped for uplink traffic.
  - APN AMBR Uplink Bytes Drop: Indicates the number of APN-AMBR bytes that are dropped for uplink traffic.
  - APN AMBR Uplink Pkts IP pref lowered: Indicates the number of APN-AMBR uplink packets for which IP precedence is lowered.
  - APN AMBR Uplink Bytes IP pref lowered: Indicates the number of APN-AMBR uplink bytes for which IP precedence is lowered.
  - APN AMBR Downlink Pkts Drop: Indicates the number of APN-AMBR packets that are dropped for downlink traffic.
  - APN AMBR Downlink Bytes Drop: Indicates the number of APN-AMBR bytes that are dropped for downlink traffic.

- APN AMBR Downlink Pkts IP pref lowered: Indicates the number of APN-AMBR downlink packets for which IP precedence is lowered.
- APN AMBR Downlink Bytes IP pref lowered: Indicates the number of APN-AMBR downlink bytes for which IP precedence is lowered.



CHAPTER 11

APN Data Tunnel MTU Size Configuration

- Revision History, on page 107
- Feature Description, on page 107
- Configuring MTU, on page 108

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The enhanced packet core (EPC) defines many different interfaces that require encapsulation of IPv4 and IPv6 data packets. Because the EPC adds encapsulating headers, additional care must be taken when fragmenting IPv4 and IPv6 packets.

Appropriate configuration should not result in fragmentation at any node in EPC. This feature fragments the IPv6 and IPv4 packets based on their MTU.

In RFC-4861 there is a provision to send the Maximum Transmission Unit (MTU) in Router Advertisement (RA) messages. P-GW supports the sending of the IPv6 MTU option in RAs for IPv6 and IPv4v6 PDN types towards the UE. The (Internet) can now send downlink data packet and based on the configured MTU, data fragmentation is performed at the source, if required. This feature also reduces the number of ICMPv6 Packet Too Big Error messages in the customer's network.

The MTU size is configurable through the Command Line Interface (CLI) on P-GW.

Limitation

- For P-GW/SAEGW IPv6 session, when packet exceeds the APN MTU value the CLI **policy ipv6 tunnel mtu exceed notify-sender** is not supported as ICMP is not available in VPP.
- For GGSN/P-GW/SAEGW IPv4 session, when packet (with df bit) exceeds the APN MTU value the CLI **access-link ip-fragmentation df-fragment-and-icmp-notify** is not supported as ICMP is not available in VPP.
- For GGSN/P-GW/SAEGW IPv4 session, when packet (with df bit) exceeds the APN MTU value the CLI **access-link ip-fragmentation normal** is not supported as ICMP is not available in VPP.

Configuring MTU

The following CLI commands configures the Maximum Transmission Unit (MTU) for data sent on the IPv4 and IPv6 tunnel between the P-GW and the mobile node:

```
configure
   context context_name
      apn apn_name
         ppp mtu bytes
         data-tunnel mtu bytes
         policy ipv6 tunnel mtu exceed { fragment inner | notify-sender |
fragment }
         access-link ip-fragmentation { df-ignore  | normal |
df-fragment-and-icmp-notify }
         end
```

NOTES:

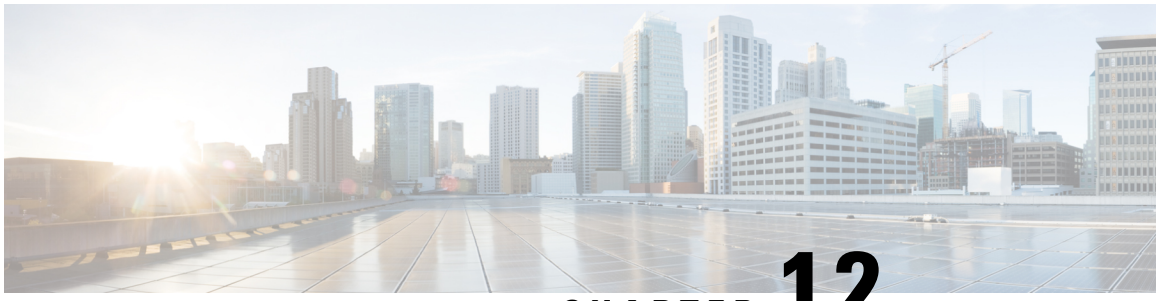
- *bytes*: Specifies the MTU for the IPv6 tunnel between the P-GW and the mobile node. bytes must be an integer between 1280 and 2000. Default: 1500.
- **ppp**: Specifies data sent on the IPv4 tunnel between P-GW and mobile node.
- **data-tunel mtu**: Specifies data sent on the IPv6 tunnel between P-GW and mobile node.
- **fragment inner**: Performs one time fragment at GTP tunnel initiator.
- **notify-sender**: System will drop the incoming packet and send "ICMPv6 Packet Too Big" to the original sender.

Note This is also the default CLI configuration, hence this should be the default behavior when nothing is explicitly configured.

- **fragment**: Performs fragmentation or reassembly at intermediate GTP hops.
- **df-ignore**: Ignores the DF (Don't Fragment) bit setting; fragments and forwards the packet over the access link.

Note This is also the default CLI configuration, hence this should be the default behavior when nothing is explicitly configured.

- **df-fragment-and-icmp-notify**: Partially ignores the DF bit; fragments and forwards the packet, but also returns an ICMP error message to the source of the packet. The number of ICMP errors sent like this is rate-limited to one ICMP error packet per second per session.
- **normal**: Drops the packet and sends an ICMP unreachable message to the source of packet.



CHAPTER 12

App-based Tethering Detection in User Plane

- Revision History, on page 111
- Feature Description, on page 111
- Configuring App-based Tethering Detection, on page 112
- Monitoring and Troubleshooting the App-based Tethering Detection, on page 113

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Important The App-based Tethering Detection is an existing feature that is supported in non-CUPS architecture. With this release, the feature is supported in the CUPS architecture.

The App-based Tethering Detection solution is built around the existing ADC plugins for App identifications. Tethering-specific patterns are added on top of recognized App plugins. These plugins successively return if the App flow is tethered or not. The App based Tethering Detection interworks with other existing supported tethering technique.

Similar to non-CUPS architecture, the tethering detection is currently supported only for Netflix and Youtube.

This feature on CUPS is in parity with non-CUPS tethering pattern detection technique.

For more information about App-based Tethering Detection, refer the *App-based Tethering Detection* chapter in the *ADC Administration Guide*.

Limitation

This feature on CUPS is in parity with non-CUPS tethering pattern detection technique. And so, if there are any new TLS patterns used by tethered devices in the network, then those are not identified for tethering detection.

Configuring App-based Tethering Detection

This section describes how to enable support for App-based Tethering Detection.

Enabling App-based Tethering Detection at Rulebase Level

Important The tethering configuration must be done on Control Plane and then, it must be pushed to User Plane.

Use the following commands to enable App-based Tethering Detection for ADC traffic under ACS Rulebase Configuration Mode:

```
configure
  active-charging service service_name
    rulebase rulebase_name
      tethering-detection application
      exit
    exit
  exit
```

NOTES:

- The **default tethering-detection** command configures its default setting.

  Default: By default, the Tethering Detection feature is disabled.

Important The OS and UA-based tethering detection are currently not supported in CUPS.

- If previously configured, use the **no tethering-detection** command to remove the tethering detection configuration from the rulebase.

Enabling App-based Tethering Detection at Ruledef Level

Use the following configurations to enable App-based Tethering Detection at Ruledef Configuration mode:

```
configure
  active-charging service service_name
    ruledef ruledef_name
     tethering-detection application { flow-tethered | flow-not-tethered
 }
      exit
```

```
    exit
  exit
```

NOTES:

- If previously configured, use the **no tethering-detection** command to remove the tethering detection configuration from the ruledef.

Monitoring and Troubleshooting the App-based Tethering Detection

This section provides information regarding commands available to monitor and troubleshoot the App-based Tethering Detection.

Show Commands and Outputs

This section provides information regarding show commands and their outputs in support of this feature.

show user-plane-service statistic tethering-detection

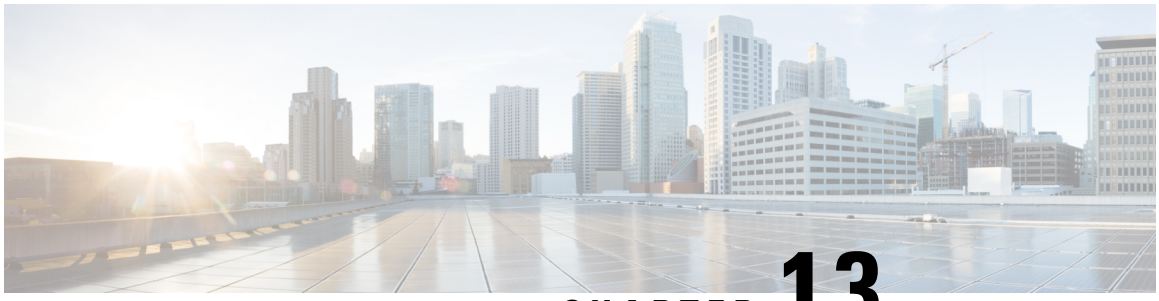
The output of this CLI command has been enhanced to include the following fields in support of this feature.

- Tethering Detection Statistics (Application):
  - Total flows scanned
  - Tethered flows detected
  - Tethered uplink packets
  - Tethered downlink packets

show user-plane-service statistic rulebase name <rulebase_name>

The output of this CLI command has been enhanced to include the following fields in support of this feature.

- Tethering Detection (Application):
  - Total flows scanned
  - Tethered flows detected
  - Tethered uplink packets
  - Tethered downlink packets



CHAPTER 13

APN and APN Profile-Based User Plane Selection for CUPS

- Revision History, on page 115
- Feature Description, on page 115
- How It Works, on page 115
- Configuring APN-Based UP Grouping, on page 118
- Monitoring and Troubleshooting APN-Based UP Grouping, on page 120

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Currently, in CUPS architecture, User Plane is selected by SAEGW-C using an algorithm that selects least connected User-Plane. Also, there is a flat list of User Plane among which User Plane is selected.

With this feature, Operator can select User Plane from specific UP group associated with an APN/APN Profile.

How It Works

Cisco CUPS solution supports Static UP selection. This is based on static selection of active SAEGW-U available. UP Group concept is used for static UP selection. UP group is a group of UP SAEGW-Us. Each APN is then associated with one UP group. APN is served by the UP groups associated with it. UPs are selected using an algorithm that selects least connected User-Plane available in that particular group.

UP Group

On SAEGW-C, for Static UP selection, UP group concept is used. An UP group is a list of UPs (SAEGW-Us). An UP can be part of only one UP group. In a UP Group, all the UPs need to be of same capacity and capability. Different type of UPs should be part of different UP group. The following two types of UP groups are supported:

- Specific UP group—It is a set of explicitly configured UPs. Specific group gives the flexibility to group certain specific types of UPs together. This helps in reserving specific set of UPs for a specific purpose. There can be multiple Specific groups that can be configured.
- Default UP group—This is a default group which groups all the UPs that are registered and are not explicitly configured as part of any Specific UP group. Default group has advantage of registering UPs in zero touch manner without having the need to explicitly configure a UP on the CP. This kind of group is more suited for collocated CUPS case where all the UPs are of same capacity and capability and are in the same data center. Default group optimizes the UP config on the CP.

An APN can be associated with UP group. If no group is associated with an APN, then default UP group is used to serve that APN. Similarly, for selecting UP for Pure-S calls, UP group can be associated to an APN profile. If there is no APN Profile/Operator-Policy defined or no group is associated with APN Profile, then SAEGW-C uses “default” UP group for selection.

Operator can reserve certain UPs for certain application. For example, IMS vs. Internet vs. IOT can have different UP groups.

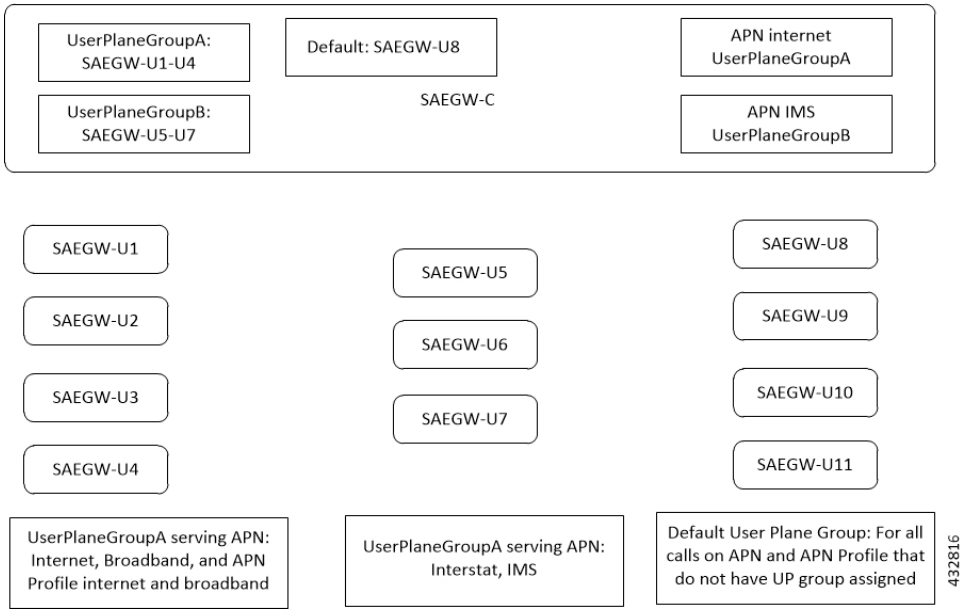
With this feature:

- SAEGW-C always has one User Plane group with the name “default”.
- SAEGW-C supports maximum of 100 User Planes.
- User planes can be organized in different group.
- Currently, 100 user-plane-groups can be configured, and single group can have maximum of 100 User Planes.
- One User Plane can be part of only one User Plane group.
- Operator can configure multiple User Planes in specific user-plane-group and in default group.
- User Planes which are associated with SAEGW-C but are not defined in any User Plane group are added in default group.
- Operator can associate User Plane Group to APN and APN Profile.
- If there is no User Plane group associated to an APN then for Pure-P and Collapsed call, the SAEGW-C uses default group to select User Plane for that session.
- If there is no User Plane group associated to APN Profile or no APN Profile is defined, then for Pure-S calls SAEGW-C uses “default” user plane group.
- For PGW Multi-PDN call, same User Plane is selected.
- User Plane group associated with APN is also used while sending IP Pool chunks to User Plane. IP Pool associated with APN is broken down to chunks and are available for distribution to all UPs from group associated with APN.
- For user-plane-groups which are not associated with any APN, SAEGW-C does not send any IP Pool chunks to UPs belonging to these group. This is applicable to default group also.

- Sessions with static IP address (IPv4/IPv6) are supported. The User Plane selection of static session is fixed as per chunk allocation to User Plane from User Plane group associated to an APN.
- If same static IP address range is used across multiple APN, then it is recommended to use same User Plane group in those APN.

Architecture

The following figure depicts a high-level architecture of this feature.



Session Recovery and ICSR

Sx-Demux Recovery, ICSR and Sessmgr and VPNmgr recovery is supported

Limitations

In CUPS architecture, this feature has the following known limitations:

- SAEGW-C does not support IPv4v6 PDN type call with static address received from UE, even if one of the IP address (either IPv4 or IPv6, or both) is static address.
- SAEGW-C does not support “allow-static” type pool configuration.
- Multi-PDN call with static IP address allocation is not supported.

Licensing

This feature is license-controlled. Contact your Cisco Account representative for license related details.

Configuring APN-Based UP Grouping

This section provides information about configurations available in support of this feature.

Prerequisites:

- Same IP context should be present at Control-Plane as well as in User-Plane.
- IP context name which is specified in APN configuration should be same at Control-Plane and User-Plane.

Configuring User Plane Group in Control Plane

New user-plane-group is defined at the global configuration mode which lists User Plane endpoints

1. User Plane Group name “default” is created by default. Operator can add and remove peer-node-id in default group. Operator cannot delete user-plane-group “default
2. If Sx Association Setup Request is received for any User plane node-id which is not part of any defined User Plane Group, it will be part of Default User Plane Group.

Configuring User Plane Group

Use the following CLI commands to configure User Plane endpoint group in Control Plane.

```
configure
   [ no ] user-plane-groupgroup_name
   end
```

Notes:

- Removal of user-plane-group will trigger Sx-Association release from Control Plane of individual peer id from that group.

Configuring Peer Node ID and User Plane Node IP Address

Use the following configuration commands to configure time-based PCC rule.

```
configure
   user-plane-group group_name
      [ no ] peer-node-id { ipv4-address | ipv6-address }
      end
```

Notes:

- Removal of peer-node-id will trigger Sx-Association Release from Control Plane for that peer id.

Verifying the User Plane Group

Use the following CLI command for verification.

```
show user-plane-group { all | name group_name }
```

Associating User Plane Group with APN

It is desired that calls to a particular APN be connected to a certain group of user-planes based on some predefined selection criteria. Operator can associate User Plane Group to APN Configuration.

User Plane group configured to APN is also used while sending IP Pool chunks to User Plane. If there is IP Pool associated with APN, only then the chunks from that pool will be sent to all User Planes in this group.

User Plane Group configuration in APN is used to select User Plane for P-GW Pure-P and Collapsed Call.

If there is no specific group is configured in APN then “default” group will be used.

Configuring User Plane Group in APN

Use the following CLI commands to configure User Plane group in APN.

```
configure
  context context_name
    apn apn_name
      [ no ] user-plane-group group_name
      end
```

NOTE: In this EFT release, removal or change of user-plane-group from APN is not supported.

Verifying the User Plane Group in APN

Use the following CLI command for verification.

```
show apn name apn_name }
```

Method of Procedure (MOP) to Remove or Change User Plane Group from APN

When explicit user-plane-group is configured, or implicit default group is used, the SAEGW-C sends IP Pool chunks from the pool that is configured (or global pool when there is no explicit pool configuration in APN) to the user planes in the group.

If you want to change or remove user-plane-group associated to a APN, then it is recommended to follow this MOP because, currently, there is no support of run time config change of user-plane-group in APN after User Plane is associated with SAEGW-C.

Before changing user-plane-group in APN it is recommended to use the following CLI command to first gracefully clear all existing calls belonging to user-plane-group associated with APN.

```
clear subscribers saegw-only user-plane-group group_name no-select-up
```

Executing this CLI command releases all sessions from User Plane belonging to the mentioned user-plane-group gracefully, and marks that User Plane as "Not Available for Session Selection". This User Plane continues to be in Associated state, but it will not be available for Session selection.

Note When the **clear subscribers** command is executed on UP, CP will not be informed and CP will consider the sessions as running.

After clearing the session, execute either of the following CLI command on User Plane to remove its association from Control Plane, and make required changes after UP association is released.

```
no user-plane-service service_name
```

Or:

```
no peer-node-id { ipv4-address ipv4_address | ipv6-address ipv6_address}
```

Associating User Plane Group with APN Profile

To select User Plane for S-GW Pure-S calls, SAEGW-C uses user-plane-group associated with APN Profile under Operator Policy. When APN profile do not have any user-plane-group associated or no APN profile was used, then SAEGW-C will select User Plane from default user-plane-group.

Configuring User Plane Group in APN Profile

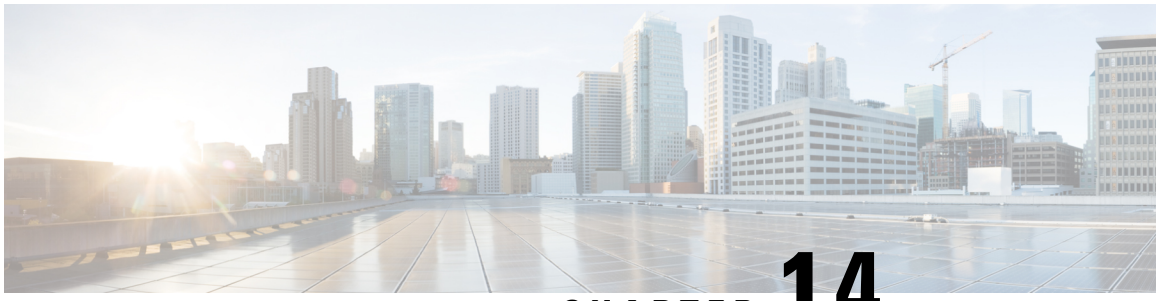
Use the following CLI commands to configure User Plane group in APN.

```
configure
  apn-profile profile_name
    [ no ] user-plane-group group_name
    end
```

Monitoring and Troubleshooting APN-Based UP Grouping

The output of the following CLI commands has been enhanced in support of this feature.

- **show sx peers**
  - Group Name Column in the output of this command is the name of user-plane-group under which Peer is configured at Control Plane.
  - Peer, which is not part of any group, will be added under user-plane-group “default”
  - For user-plane-group which is not associated with any “apn”, SAEGW-C will not send any IP Pools to User Planes from this group. And so, in the output of this command, the Group Name that is not associated with and “apn”, the IP Pool status will be “N – Not Applicable”. Also, for User Planes in this group, when “show sx peers” is executed on User Plane, it will show Peer ID as “0”.
- **show ip user-plane**
- **show ip pool-chunks up-id** *up_id* **user-plane-group name** *up_group_name*



CHAPTER 14

Cisco Ultra Traffic Optimization with VPP

- Revision History, on page 121
- Feature Description, on page 121
- RCM Support, on page 122
- Sending the GBR or MBR Values to Cisco Ultra Traffic Optimization , on page 122
- How it Works, on page 123
- Show Commands and Outputs, on page 124
- Sample Configuration, on page 130

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Cisco Ultra Traffic Optimization is supported on VPP in the CUPS architecture.

The Cisco Ultra Traffic Optimization is a RAN optimization technology that increases subscriber connection speeds in congested cells and, as a result, increases the cell capacity significantly. The result is an optimized RAN, where Mobile Network Operators (MNOs) can deploy fewer cells, on an ongoing basis, and absorb more traffic growth while meeting network quality targets.

Large traffic flows, such as Adaptive Bit Rate (ABR) video, saturate radio resources and swamp the eNodeB scheduler. The Cisco Ultra Traffic Optimization employs machine learning algorithms to detect large traffic flows (such as video) in the network and optimize the delivery of those flows to mitigate the network congestion without changing user quality (that is, video works the same for the end user). In other words, by employing software intelligence at the network core, Cisco Ultra Traffic Optimization mitigates the overwhelming impact video has on the RAN.

The resulting benefits are seen in congested network sites. The Cisco Ultra Traffic Optimization:

- Increases average user throughput.
- Increases congested cell site capacity.
- Reduces scheduler latency.
- Maintains user quality of experience even when more users and more traffic share a cell.
- Is measured directly by eNodeB performance counters (for example, average UE throughput, scheduler latency), which are the key performance indicators that are used for network capacity planning.
- Provides permanent savings in RAN investment requirements.
- Is integrated in the Cisco StarOS P-GW.
- Requires no new hardware or cabling complexity - it can be turned on for a market in an hour.
- Supports HTTP(s) and QUIC traffic.

Licensing

The Cisco Ultra Traffic Optimization with VPP is a licensed Cisco solution. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

RCM Support

This feature enables the Redundancy and Configuration Management (RCM) support for the Cisco Ultra Traffic Optimization (CUTO). All relevant configuration to enable the Cisco Ultra Traffic Optimization (CUTO) using service scheme and application of the Cisco Ultra Traffic Optimization (CUTO) profile or policy on User Plane is supported using RCM.

Sending the GBR or MBR Values to Cisco Ultra Traffic Optimization

During the stream create/update, a bearer with valid QER and is GBR bearer, the respective bearer level downlink GBR/MBR values are sent to Cisco Ultra Traffic Optimization (CUTO) library as lower or upper limit values otherwise lower limit or upper limit values are zero. The values of lower limit and upper limit are in Bits Per Second (BPS). Post RCM support, the P-GW sends the downlink flow level GBR and MBR values instead of bearer level GBR and MBR to the optimization library. For GBR bearer, flow level GBR is sent as lower limit and flow level MBR is sent as the upper limit to the Cisco Ultra Traffic Optimization (CUTO) library. For non-GBR bearer 0 is sent as lower limit and flow level MBR is sent as upper limit to the Cisco Ultra Traffic Optimization (CUTO) library. If the flow level MBR is greater than the APN-AMBR for a non GBR bearer, traffic is throttled at APN-AMBR. In such a case APN-AMBR is sent as the upper limit to the Cisco Ultra Traffic Optimization (CUTO) library. If there is no valid flow level MBR specific to the flow, APN-AMBR is sent as the upper limit to the Cisco Ultra Traffic Optimization (CUTO) library. Optimization library maintains logical flow based on 3-tuple (That is source IP, destination IP and protocol), whereas the non-CUPS architecture considers a flow as 5-tuple (That is source IP, destination IP, source port, destination port and protocol). Hence multiple non-CUPS architecture 5-tuple entries can belong to same

3-tuple entry in optimization library. The PG-W provides GBR and MBR values based on 5-tupple to the optimization library. As part of this feature:

- Optimization library uses the minimum of all MBR values that belong to same 3-tuple entry as upper limit.
- Optimization library uses maximum of all GBR values that belong to same 3-tuple entry as lower limit.

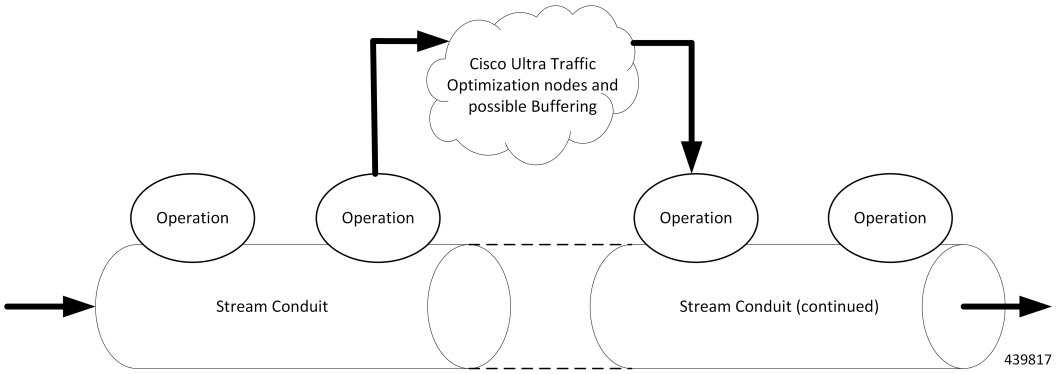
Cisco Ultra Traffic Optimization Library Deinitialization

This feature currently doesn't support the Deinitialization. Deinitialization happens when the Cisco Ultra Traffic Optimization (CUTO) license is removed from the system.

How it Works

Architecture

The following figure illustrates the architecture of Cisco Ultra Traffic Optimization on VPP in CUPS.



Cisco Ultra Traffic Optimization is split across Control Plane and User Plane.

CUTO-CTRL

- CUTO-CTRL receives guidance and requests from SMGR through the East-West API (EWAPI), through which clients (SMGR instances) are registered and de-registered, and new streams/flows are created and terminated.
- CUTO-CTRL manages a set of shared memory (SHM) tables using a North-South API (NSAPI) consisting of Cisco-provided SHM infrastructure.
- It is through this SHM environment that CUTO-VPP can read and write content that is visible to both CUTO-VPP and CUTO-CTRL.
- The SHM is used for all high volume, scalable/mutable content necessary for the high-performance configuration and administration of the CUTO solution in VPP.

CUTO-VPP

- CUTO-VPP is the packet processing engine in the user plane.
- In fastpath, Cisco Ultra Traffic Optimization is applied to packets on a stream configured with its operation.
- Packets are sent from the Stream conduit to a particular CUTO-VPP operation, and after some potential delay (0-N milliseconds), traffic is returned to the same Conduit.
- Packets are never dropped by the Cisco Ultra Traffic optimization application.

Limitations

The Cisco Ultra Traffic Optimization feature in CUPS has the following limitations:

- CUTO configuration changes done in Service Schema do not take effect immediately for existing flows.
- Cisco Ultra Traffic Optimization VPP global deinitialization is not supported.
- Dynamic memory allocation between SMGR and CUTO-VPP.
- Bearer-related triggers for enabling Cisco Ultra Traffic Optimization are not supported.
- Rule match change trigger must be configured for CUTO in CUPS.
- Disabling of Traffic optimization is not supported on 'loc-update' trigger.
- Enabling Cisco Ultra Traffic Optimization via Gx is not supported.
- Removal of CUTO license will not trigger global deinitialization. CUTO configurations must be removed to disengage CUTO functionality for new flows.

Show Commands and Outputs

This section provides information regarding show commands and their outputs in support of Cisco Ultra Traffic Optimization in CUPS.

For information on other supporting show commands, refer to *Monitoring and Troubleshooting* section under the *Cisco Ultra Traffic Optimization* chapter in the *P-GW Administration Guide*.

Show Commands and Outputs

show user-plane-service traffic-optimization counters sessmgr all

The output of this command includes the following fields:

TCP Traffic Optimization Flows:

- Active Normal Flow Count
- Active Large Flow Count
- Active Managed Large Flow Count
- Active Unmanaged Large Flow Count

- Total Normal Flow Count
- Total Large Flow Count
- Total Managed Large Flow Count
- Total Unmanaged Large Flow Count
- Total IO Bytes
- Total Large Flow Bytes
- Total Recovered Capacity Bytes
- Total Recovered Capacity ms

UDP Traffic Optimization Flows:

- Active Normal Flow Count
- Active Large Flow Count
- Active Managed Large Flow Count
- Active Unmanaged Large Flow Count
- Total Normal Flow Count
- Total Large Flow Count
- Total Managed Large Flow Count
- Total Unmanaged Large Flow Count
- Total IO Bytes
- Total Large Flow Bytes
- Total Recovered Capacity Bytes
- Total Recovered Capacity ms

show user-plane-service traffic-optimization info

The output of this command includes the following fields:

- CUTO Ctrl Library Version
- CUTO VPP Library Version
- Mode
- Configuration

show user-plane-service traffic-optimization policy all

The output of this command includes the following fields:

- Policy Name
- Policy-Id

- Bandwidth-Mgmt
  - Backoff-Profile
  - Min-Effective-Rate
  - Min-Flow-Control-Rate
- Curbing-Control:
  - Time
  - Rate
  - Max-Phases
  - Threshold-Rate
- Heavy-Session:
  - Threshold
  - Standard-Flow-Timeout
- Link-Profile:
  - Initial-Rate
  - Max-Rate
  - Peak-Lock
- Session-Params:
  - Tcp-Ramp-Up
  - Udp-Ramp-Up

Bulkstats

The following existing bulk statistics are supported by Cisco Ultra Traffic Optimization in CUPS:

Bulk Statistics	Description
cuto-uplink-drop	Indicates the total number of uplink packets dropped by CUTO library
cuto-uplink-hold	Indicates the total number of uplink packets held by CUTO library
cuto-uplink-forward	Indicates the total number of uplink packets forwarded by CUTO library
cuto-uplink-rx	Indicates the total number of uplink packets received by CUTO library
cuto-uplink-tx	Indicates the total number of uplink packets sent by CUTO library

Bulk Statistics	Description
cuto-dnlink-drop	Indicates the total number of downlink packets dropped by CUTO library
cuto-dnlink-hold	Indicates the total number of downlink packets held by CUTO library
cuto-dnlink-forward	Indicates the total number of downlink packets forwarded by CUTO library
cuto-dnlink-rx	Indicates the total number of downlink packets received by CUTO library
cuto-dnlink-tx	Indicates the total number of downlink packets sent by CUTO library
cuto-todrs-generated	Indicates the total number of TODRs generated.
tcp-active-normal-flow-count	Indicates the number of TCP active-normal-flow count for Cisco Ultra Traffic Optimization.
tcp-active-large-flow-count	Indicates the number of TCP active-large-flow count for Cisco Ultra Traffic Optimization.
tcp-active-managed-large-flow-count	Indicates the number of TCP active-managed-large-flow count for Cisco Ultra Traffic Optimization.
tcp-active-unmanaged-large-flow-count	Indicates the number of TCP active-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-normal-flow-count	Indicates the number of TCP total-normal-flow count for Cisco Ultra Traffic Optimization.
tcp-total-large-flow-count	Indicates the number of TCP total-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-managed-large-flow-count	Indicates the number of TCP total-managed-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-unmanaged-large-flow-count	Indicates the number of TCP total-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-io-bytes	Indicates the number of TCP total-IO bytes for Cisco Ultra Traffic Optimization.
tcp-total-large-flow-bytes	Indicates the number of TCP total-large-flow bytes for Cisco Ultra Traffic Optimization.
tcp-total-recovered-capacity-bytes	Indicates the number of TCP total-recovered capacity bytes for Cisco Ultra Traffic Optimization.
tcp-total-recovered-capacity-ms	Indicates the number of TCP total-recovered capacity ms for Cisco Ultra Traffic Optimization.

Bulk Statistics	Description
udp-active-normal-flow-count	Indicates the number of UDP active-normal-flow count for Cisco Ultra Traffic Optimization.
udp-active-large-flow-count	Indicates the number of UDP active-large-flow count for Cisco Ultra Traffic Optimization.
udp-active-managed-large-flow-count	Indicates the number of UDP active-managed-large-flow count for Cisco Ultra Traffic Optimization.
udp-active-unmanaged-large-flow-count	Indicates the number of UDP active-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-normal-flow-count	Indicates the number of UDP total-normal-flow count for Cisco Ultra Traffic Optimization.
udp-total-large-flow-count	Indicates the number of UDP total-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-managed-large-flow-count	Indicates the number of UDP total-managed-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-unmanaged-large-flow-count	Indicates the number of UDP total-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-io-bytes	Indicates the number of UDP total-IO bytes for Cisco Ultra Traffic Optimization.
udp-total-large-flow-bytes	Indicates the number of UDP total-large-flow bytes for Cisco Ultra Traffic Optimization.
udp-total-recovered-capacity-bytes	Indicates the number of UDP total-recovered capacity bytes for Cisco Ultra Traffic Optimization.
udp-total-recovered-capacity-ms	Indicates the number of UDP total-recovered capacity ms for Cisco Ultra Traffic Optimization.

The following statistics for Cisco Ultra Traffic Optimization, that are part of the legacy (StarOS) implementation, are not applicable to the CUPS implementation.

- tcp-uplink-drop
- tcp-uplink-hold
- tcp-uplink-forward
- tcp-uplink-forward-and-hold
- tcp-uplink-hold-failed
- tcp-uplink-bw-limit-flow-sent
- tcp-dnlink-drop
- tcp-dnlink-hold

- tcp-dnlink-forward
- tcp-dnlink-forward-and-hold
- tcp-dnlink-hold-failed
- tcp-dnlink-bw-limit-flow-sent
- tcp-dnlink-async-drop
- tcp-dnlink-async-hold
- tcp-dnlink-async-forward
- tcp-dnlink-async-forward-and-hold
- tcp-dnlink-async-hold-failed
- tcp-process-packet-drop
- tcp-process-packet-hold
- tcp-process-packet-forward
- tcp-process-packet-forward-failed
- tcp-process-packet-forward-and-hold
- tcp-process-packet-forward-and-hold-failed
- tcp-pkt-copy
- tcp-pkt-Copy-failed
- tcp-process-pkt-copy
- tcp-process-pkt-copy-failed
- tcp-process-pkt-no-packet-found-action-forward
- tcp-process-pkt-no-packet-found-forward-and-hold
- tcp-process-pkt-no-packet-found-action-drop
- tcp-todrs-generated
- udp-uplink-drop
- udp-uplink-hold
- udp-uplink-forward
- udp-uplink-forward-and-hold
- udp-uplink-hold-failed
- udp-uplink-bw-limit-flow-sent
- udp-dnlink-drop
- udp-dnlink-hold
- udp-dnlink-forward

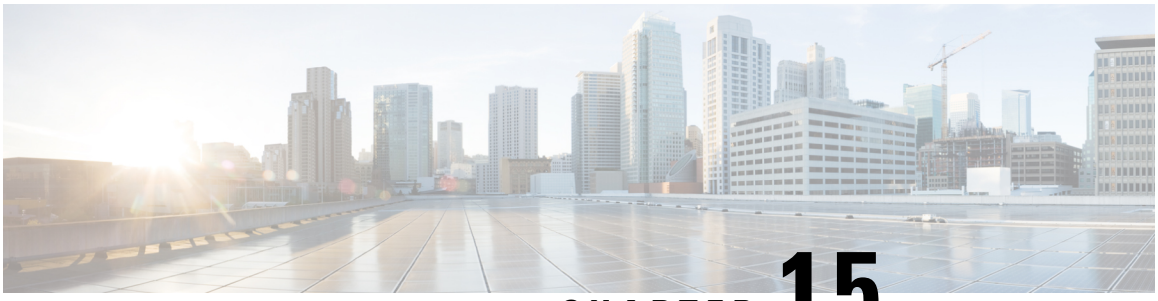
- udp-dnlink-forward-and-hold
- udp-dnlink-hold-failed
- udp-dnlink-bw-limit-flow-sent
- udp-dnlink-async-drop
- udp-dnlink-async-hold
- udp-dnlink-async-forward
- udp-dnlink-async-forward-and-hold
- udp-dnlink-async-hold-failed
- udp-process-packet-drop
- udp-process-packet-hold
- udp-process-packet-forward
- udp-process-packet-forward-failed
- udp-process-packet-forward-and-hold
- udp-process-packet-forward-and-hold-failed
- udp-pkt-copy
- udp-pkt-Copy-failed
- udp-process-pkt-copy
- udp-process-pkt-copy-failed
- udp-process-pkt-no-packet-found-action-forward
- udp-process-pkt-no-packet-found-forward-and-hold
- udp-process-pkt-no-packet-found-action-drop
- udp-todrs-generated

Sample Configuration

Sample configuration to enable CUPS CUTO feature:

```
configure
  active-charging service ACS
    trigger-action TA1
       traffic-optimization policy custom1
    #exit
    trigger-condition TC1
      rule-name = dynamic-rule2
    #exit
    service-scheme SS1
      trigger rule-match-change
        priority 5 trigger-condition TC1 trigger-action TA1
      #exit
```

```
    subs-class SB1
      rulebase = cisco
    #exit
    subscriber-base default
      priority 5 subs-class SB1 bind service-scheme SS1
    #exit
    traffic-optimization-profile
      mode active
      data-record
    #exit
    traffic-optimization-policy custom1
      bandwidth-mgmt min-effective-rate 300 min-flow-control-rate 150
      heavy-session threshold 200000
      link-profile max-rate 20000
    #exit
    traffic-optimization-policy default
    #exit
end
```



CHAPTER 15

Charging Action Configuration Change Support for Existing Sessions Gy and Gz Interface

- Revision History, on page 133
- Feature Description, on page 133
- How It Works, on page 134

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Configuration gap leads to improper usage report being sent to Gy and Gz in CDR, and delay in checkpointing also results in data loss due to the SR/ICSR process.

To overcome the gaps in the existing Pure-P and collapsed calls for Gy/Gz charging, the following configuration changes are implemented:

- “Rated to Free” and “Free to Rated” mid-call in the charging-action.
- Addition of high priority rule mid-call with different charging-action that has different Rating Group.
- Charging-action mid-call for the ruledef.

How It Works

Support for the following is added in the existing calls for configuration change related to charging-action and addition of high priority rule within the Rulebase.

Configuration Change Under Charging Action from Rated to Free and Free to Rated Mid Call

For the configuration change under the charging-action from “free to rated”, the Control Plane (CP) applies the change and creates the required URR for the Gy/Gz components. And, when the User Plane reports the usage report, the same report goes to the Gy interface and in the CDR based on the new changes.

When configuration change under charging-action is done from “rated to free”, The User Plane (UP) sends the data usage only for the rated configuration in the usage report to CP. After receiving the usage report, the same is reported to the Gy/Gz interface as applicable.

Configuration Change for Addition of High Priority Rule with Different Charging Action with Different Rating Group

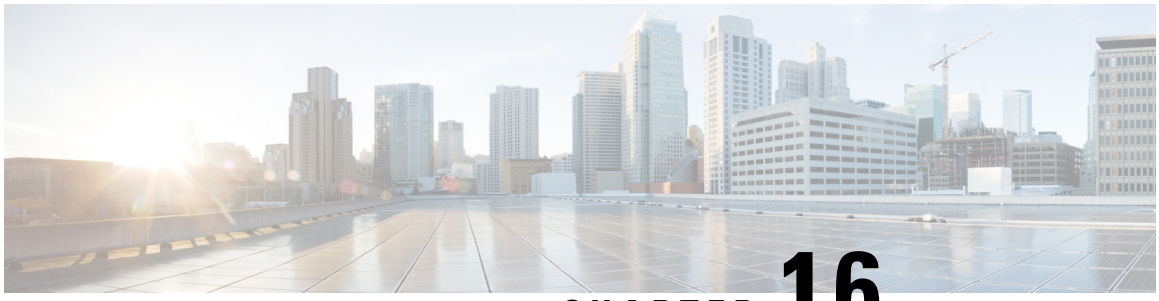
Once you apply the configuration changes in the rulebase by adding a high priority rule with different charging-action and with different rating group. The User Plane sends the separate URR for new charging-action rule and the CP compares the URR with the configuration and handles the new URR. The CP sends the corresponding information to the Gy/Gz interface as applicable.

Configuration Change for Charging-Action Mid Call for the Ruledef

When configuration change is done in the Rulebase by changing the charging-action with different charging-action and with a different rating group. The CP handles the new URR received and sends the proper LOSDV in the CDR with correct Rating Group. Also, when the User Plane sends the start of the traffic after configuration change to the CP, the CP sends the Gy to the CCR-U to request for the quota.

URR Bucket Checkpointing Enhancement for Gy

Whenever the “sx-session-usage-report” comes from User Plane to Control Plane, the checkpointing does not happen immediately. It is done as part of full checkpoint. In between, if there is session recovery, the intermediate details are missed out. To avoid this issue, the micro checkpointing is needed as and when the “sx-session-usage-report” reaches to the Control Plane.



CHAPTER 16

Dedicated Bearer Establishment without PCRF

- Revision History, on page 135
- Feature Description, on page 135
- How it Works, on page 135
- Configuring active-charging-services, on page 139

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

To provide IMS services to the UE that aren't VOLTE capable, P-GW uses the deep packet inspection (DPI) functionality to create dedicated bearers without interaction with PCRF. This helps in maintaining high QoS of the voice service although the default bearer for the internet APN gets created with interaction with PCRF.

SBC IP address (IPv4 or IPv6) and protocol RTP/RTCP are configured in ruledef and a dedicated bearer is created when a subscriber traffic matches with the ruledef without interaction with PCRF to detect voice services. If no data flows, then the dedicated bearer gets removed after the configured time limit and there's no interaction with PCRF.

How it Works

The service schema framework in CUPS supports the dedicated bearer establishment in GW when default bearer is created via PCRF. Trigger condition and trigger action are configured under service schema to create new traffic flow. For dedicated bearer establishment, the rule name configured in trigger condition must match with the rule name in the rule base configuration which is the default bearer rule name.

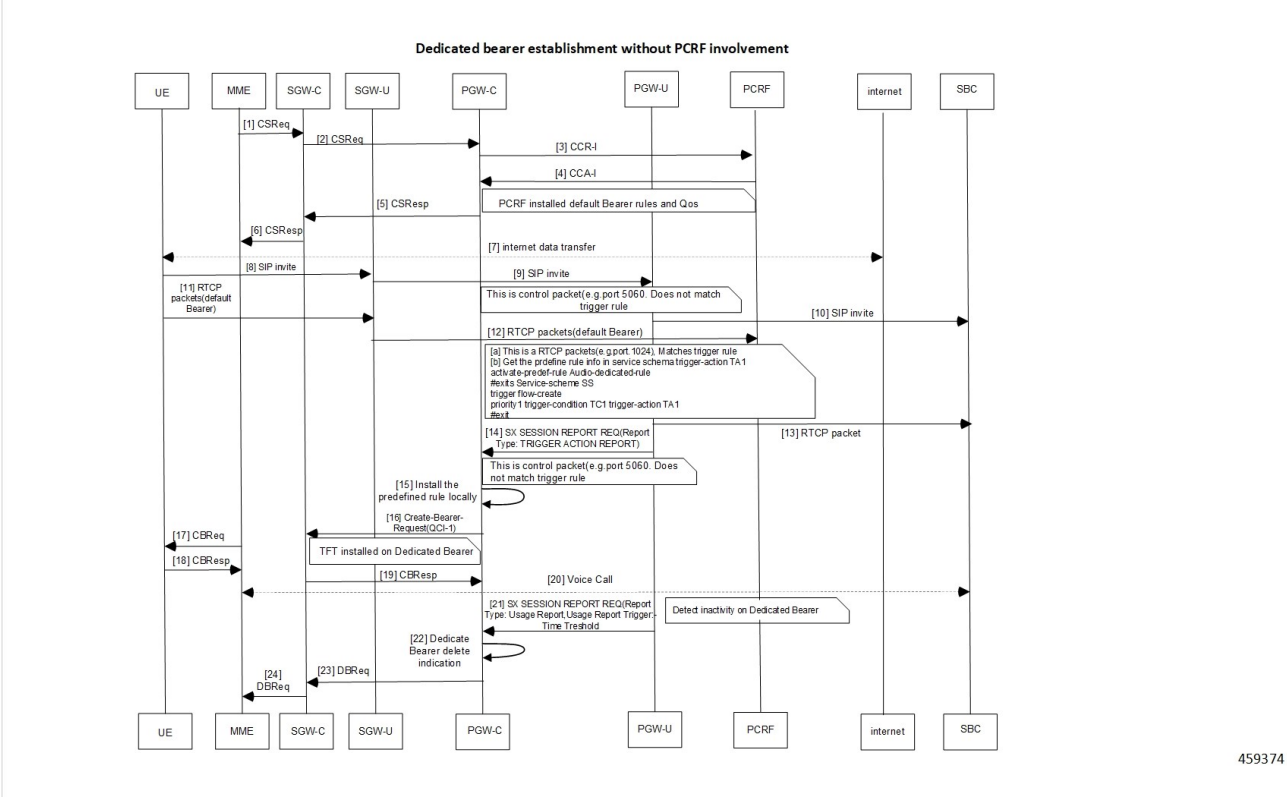
The predefined rule trigger action gets activated when the traffic matches with the corresponding IP and the configured port range in the rule. This results in a new dedicated bearer activation without an interaction with PCRF. If no data flows on this bearer after the predefined configured time limit then it gets deleted.

Some of the key highlights are:

- There's a separate APN for non-VoLTE UEs, to avoid creation attempt of a dedicated bearer for VoLTE UEs.
- If dedicated bearer creation fails, call continues and the traffic continues to flow on this bearer.
- Retransmission for SX_Session_Report_Request intended for dedicated bearer creation works as per existing behavior.
- Feature support is provided for P-GW/SAEGW CUPS calls only. GGSN CUPS doesn't support dedicated bearer in this release.

This section describes call flow and procedure on how the dedicated bearer is established without involving PCRF.

Figure 7: Call Flow



459374

Table 1: Procedure

Step	Description
1	Establish default bearer with internet APN with PCRF.

Step	Description
2	Default Bearer receives a SIP invite and forwards to SBC. **Note** The port 5060/5061 receives the SIP invite, which is outside the port range for dedicated bearer.
3	An RTCP packet is received matching the SBC address and port-range that matches with the trigger rule.
4	After the rule match, UP reads the service schema and identifies the predefined rule information that is required for the creation of a dedicated bearer.
5	UP shares the rule information through **SX_Session_Report_Request** (for report type Trigger Action Report) with CP. CP process the request message and triggers the rule installation.
6	A dedicated bearer is created with predefined **Audio_dedicated_rule** and TFT with multiple ranges of port as defined by packet-filters tft1 and tft2. **Note** The charging-action for such rule has **billing-action egcdr** and **content-id** configured.
7	UE pushes any packets for the specified port ranges to the dedicated bearer and matches with the Audio_dedicated_rule.
8	After 300 seconds or configured timeout value of inactivity according to the threshold configuration, the bearer gets deleted and PGWCDR gets generated for a bearer with corresponding data counts. • You can adjust the threshold based on the keep-alive/watchdog messages. • The SIP control messages after voice call flows on the ports that are outside the port-range defined for dedicated bearer TFT. **Note** As UE sends SIP control messages on the default bearer, it's not considered under dedicated bearer activity. **AF-Charging-Id** isn't populated in dedicated bearer PGWCDR.

Sx Interface Changes

During the Post rule match activity at UP, the service schema is checked to get the predefined rule information that is required for the dedicated bearer creation. This information is shared with CP through SX_Session_Report_ Request message enabling the CP to trigger the installation of the rule for dedicated bearer creation.

The rule information sent in SX_Session_Report_ Request through a newly introduced session report type "TRIGGER ACTION REPORT” and SX private IE as explained below.

Table 2: Session Report Type IE

		Bits								
	Octets	8	7	6	5	4	3	2	1	
	1 to 2	Type = 39 (decimal)								
	3 to 4	Length = n								
	5	GTER	SRIR	Spare	SPTIR	UPIR	ERIR	USAR	DLDR	
	6	Spare				TAR	NBUR	UPRR	STS	
	7 to (n+4)	These octet(s) is/are present only if explicitly specified								

Octet 6 (present when Length>1) is encoded as follows:

- Bit 1 – STS (Subscriber Trace Status Report): When set to 1, it indicates Subscriber Trace Status Report.
- Bit 2 – UPRR.
- Bit 3 – NBUR.
- Bit 4 – TAR (Trigger Action Report): When set to 1, it indicates Trigger Action Report IE.
- Bit 5 to 8 – Spare.

Trigger Action Report IE (Private IE)

This is a conditional IE applicable only for Pure-P and Collapse call types.

Table 3: Trigger Action Report IE

.		Bits								
	Octets	8	7	6	5	4	3	2	1	
	1 to 2	Type = 256 (decimal)								
	3 to 4	Length = n								
	5 to n+2	Trigger Actions								

Multiple Trigger Action IE is specified in the TAR IE. Currently, only one Trigger Action Type is packed within the Trigger Actions.

Trigger Actions

It is encoded per the following format:

Table 4: Trigger Actions

.		Bits	

	Octets	8	7	6	5	4	3	2	1	
	1	Trigger Action Type								
	2 to 3	Length = p								
	4 to (4+p)	Trigger Action Blob								

Trigger Action Type: Current value allowed = 1 (Rule Activate). It can be extended for different trigger action types in future.

Trigger Action Blob: It is unique as per trigger action type. For trigger action type = Activate Rule, it is:

Table 5: Trigger Action Blob

.		Bits								
	Octets	8	7	6	5	4	3	2	1	
	1 to p	Rule Name								

N-1 Compatibility Matrix

The following details are part of the N-1 compatibility matrix:

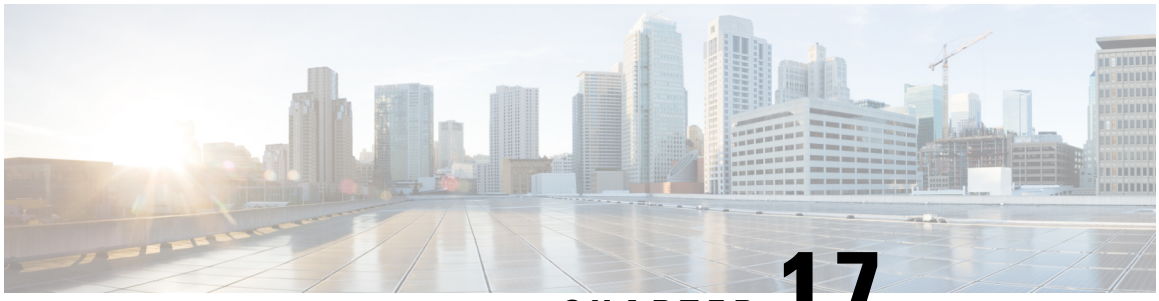
Sl.No	CP – UP	Behavior
1	CP and UP are on same version	SX_Session_Report_Request is handled at CP and trigger action is performed. CP must validate IE and return reject with Offending IE in case the TAR IE is not packed accurately.
2	CP is of older version (doesn't understand TAR in Sx Session Report Req)	UP sends the SX_Session_Report_Request with TAR bit. CP ignores this SX_Session_Report_Request and sends as success to UP.
3	Newer CP version	Older version of UP never triggers SX_Session_Report with TAR bit = 1, so no handling is needed. However, CP must validate the IE and reject with Offending IE in case the TAR IE is not packed accurately.

Configuring active-charging-services

Use the following example configuration to establish a dedicated bearer without interaction with PCRF.

```
config
  active-charging service acs
    ruledef Audio_dedicated_rule
      ip dst-address = 209.165.200.224/27
    #exit
    ruledef trigger_rule
      ip dst-address = 209.165.200.224/27
      udp either-port range 1024 to 5059
```

```
      udp either-port range 5062 to 43672
    #exit
    packet-filter tft1
      ip remote-port range 1024 to 5059
      ip remote-address = 209.165.200.224/27
    #exit
    packet-filter tft2
      ip remote-port range 5062 to 43672
      ip remote-address = 209.165.200.224/27
    #exit
    charging-action no_charge
    #exit
    charging-action ca_audio
      content-id 2
      billing-action egcdr
      qos-class-identifier 1
      flow limit-for-bandwidth direction downlink peak-data-rate 256000 peak-burst-size
32000 violate-action discard
     flow limit-for-bandwidth direction uplink peak-data-rate 256000 peak-burst-size 300000
 violate-action discard
      allocation-retention-priority 4 pvi 1 pci 1
      tft packet-filter tft1
      tft packet-filter tft2
    #exit
    rulebase prepaid
       billing-records egcdr
  #Install Audio_dedicated_rule on dedicated bearer to cater to VoLTE traffic
      action priority 1 dynamic-only ruledef Audio_dedicated_rule charging-action ca_audio
  #Use traffic matching to trigger_rule on default bearer as trigger condition
action priority 2 ruledef trigger_rule charging-action no_charge
    #exit
    trigger-action TA1
      #activate-predef-rule Audio_dedicated_rule
    #exit
    trigger-condition TC1
      rule-name = trigger_rule
    #exit
    trigger-condition tc
      rulebase = prepaid
    #exit
    service-scheme SS
      trigger flow-create
        priority 1 trigger-condition TC1 trigger-action TA1
      #exit
    subs-class SC1
      rulebase = prepaid
    #exit
    subscriber-base sb
      priority 1 subs-class SC1 bind service-scheme SS
    #exit
  #exit
context egress
    apn internet
 #Remove dedicated bearer after 300 seconds of inactivity
      timeout bearer-inactivity gbr 300 volume-threshold total 1
 active-charging rulebase prepaid
    exit
  exit
end
```



CHAPTER 17

Disable Radius Accounting

- Revision History, on page 141
- Feature Description, on page 141
- Configuring RADIUS Accounting on Dedicated Bearer Feature, on page 142

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

RADIUS is a distributed client or server system that secures networks against unauthorized access. In the Cisco implementation, the RADIUS clients run on Cisco devices and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

CUPS supports disabling RADIUS accounting on dedicated bearers for RADIUS server.

CUPS supports the following functionality:

- Enabling RADIUS accounting for all bearers
- Disabling RADIUS accounting for a specific dedicated bearer based on its QCI and ARP value
- Enabling RADIUS accounting only for the default bearer while disabling RADIUS accounting for all the dedicated bearers
- URRs are not created for bearers that have their RADIUS accounting disabled
- CLI configuration changes apply only to new calls made after the configuration change and do not affect existing calls.

- If the RADIUS accounting for a particular bearer is disabled or enabled, it applies to bearers created after it was disabled or enabled, and not for existing bearers

NOTE: This functionality is also available for products that use RADIUS in non-CUPS architecture.

Configuring RADIUS Accounting on Dedicated Bearer Feature

This section describes the CLI configurations for:

- Enabling RADIUS accounting for all bearers
- Disabling RADIUS accounting for a specific dedicated bearer based on its QCI and ARP value
- Enabling RADIUS accounting only for the default bearer while disabling RADIUS accounting for all the dedicated bearers

Enabling RADIUS Accounting for All Bearers

To enable RADIUS accounting for all the bearers, use the following CLI configuration.

```
configure
   context context_name
      aaa group group_name
         radius accounting mode all-bearers
         end
```

NOTES:

- The **radius accounting mode all-bearers** CLI command is enabled by default.

Disabling RADIUS Accounting for a Specific Bearer

To disable RADIUS accounting for a specific dedicated bearer based on its QCI and ARP values, use the following CLI configuration.

```
configure
   context context_name
      aaa group group_name
         radius accounting disable-bearer qci qci_value arp-priority-level
 arp_value
         end
```

NOTES:

- The **radius accounting disable-bearer qci** *qci_value* **arp-priority-level** *arp_value* CLI command disables RADIUS accounting only for the dedicated bearer with the specified QCI and ARP values. Accounting of other dedicated bearers is not affected.
- The maximum number of QCI and ARP combination configurations allowed to disable RADIUS accounting on dedicated bearers is 16. If you try to configure more than 16 combinations, the following error message is displayed:

```
Failure: Error!!! Maximum 16 qci and arp combinations allowed.
```

Enabling RADIUS Accounting only for the Default Bearer

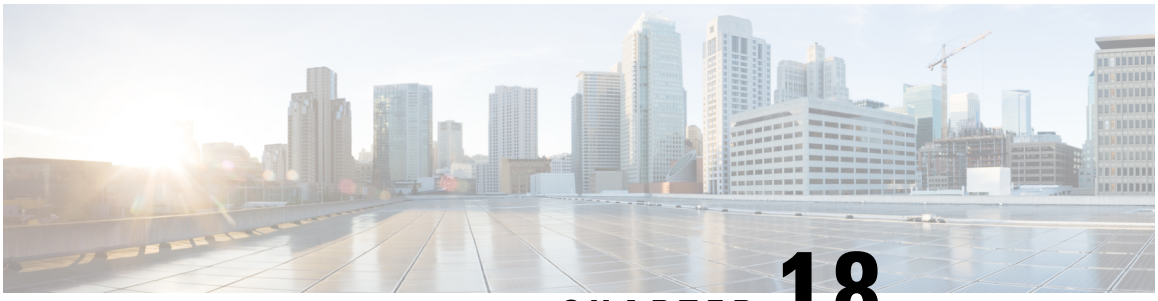
To enable RADIUS accounting only for the default bearer, and disable RADIUS accounting for all the dedicated bearers, use the following CLI configuration.

```
configure
   context context_name
      aaa group group_name
         radius accounting mode default-bearer-only
         end
```

NOTES:

- The **radius accounting mode default-bearer-only** CLI command enables RADIUS accounting only for the default bearer and disables RADIUS accounting for all the dedicated bearers.
- To remove the **radius accounting disable-bearer qci** *qci_value* **arp-priority-level** *arp_value* configuration for a specific dedicated bearer, and allow RADIUS accounting for that dedicated bearer, use the **no radius accounting disable-bearer qci** *qci_value* **arp-priority-level** *arp_value* CLI command.
- When RADIUS accounting mode is set to default-bearer-only, you cannot disable RADIUS accounting on a dedicated bearer. If you run the **radius accounting disable-bearer qci** *qci_value* **arp-priority-level** *arp_value* CLI command, the following error message is displayed:

```
Failure: Error!!! Radius accounting mode is set to default-bearer-only. Change the mode
 to all-bearers and run this CLI again
```



CHAPTER 18

Default and Dedicated Bearer Support for Pure-P and Collapsed Sessions

- Revision History, on page 145
- Feature Description, on page 145

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Provides a foundation for contributing towards improved Quality of User Experience (QoE) by enabling deterministic end to-end forwarding and scheduling treatments for different services or classes of applications pursuant to their requirements for committed bandwidth resources, jitter and delay. In this way, each application receives the service treatment that users expect.

The Cisco EPC core platforms support one or more EPS bearers (default plus dedicated). An EPS bearer is a logical aggregate of one or more Service Data Flows (SDFs), running between a UE and a P-GW in the case of a GTP-based S5/S8 interface, and between a UE and HSGW (HRPD Serving Gateway) in case of a PMIP-based S2a interface. In networks where GTP is used as the S5/S8 protocol, the EPS bearer constitutes a concatenation of a radio bearer, S1-U bearer and an S5/S8 bearer anchored on the P-GW. In cases where PMIPv6 is used the EPS bearer is concatenated between the UE and HSGW with IP connectivity between the HSGW and P-GW.

An EPS bearer uniquely identifies traffic flows that receive a common QoS treatment between a UE and P-GW in the GTP-based S5/S8 design, and between a UE and HSGW in the PMIPv6 S2a approach. If different QoS scheduling priorities are required between Service Data Flows, they should be assigned to separate EPS

bearers. Packet filters are signaled in the NAS procedures and associated with a unique packet filter identifier on a per-PDN connection basis.

One EPS bearer is established when the UE connects to a PDN, and that remains established throughout the lifetime of the PDN connection to provide the UE with always-on IP connectivity to that PDN. That bearer is referred to as the default bearer. A PDN connection represents a traffic flow aggregate between a mobile access terminal and an external Packet Data Network (PDN) such as an IMS network, a walled garden application cloud or a back-end enterprise network. Any additional EPS bearer that is established to the same PDN is referred to as a dedicated bearer. The EPS bearer Traffic Flow Template (TFT) is the set of all 5-tuple packet filters associated with a given EPS bearer. The EPC core elements assign a separate bearer ID for each established EPS bearer. At a given time a UE may have multiple PDN connections on one or more P-GWs.

With this feature, UDP, TCP, and HTTP data is offloaded to Fastpath for Default and Dedicated bearer.

Supported Functionality

For Pure-P and Collapsed session, the:

1. Default bearer establishment includes (CCA-I):
   - Default bearer establishment with and without rule.
   - Predefined Rules/Group-of-Ruledefs (GoRs).
2. Default bearer updation includes (CCA-U/RAR):
   - New Rule installation.
   - Modification of existing rules (TFT change, MBR/GBR change, Flow Status change).
   - Removal of existing rules.
   - Default Bearer QoS change.
   - APN-AMBR change.
   - Predefined Rules/GoRs.
3. Default bearer deletion includes (CCA-U/RAR):
   - Removal of existing rules.
4. Dedicated bearer establishment includes (CCA-I/CCA-U/RAR):
   - New dedicated bearer establishment.
   - Predefined Rules/GoRs.
5. Dedicated bearer updation includes (CCA-U/RAR):
   - Addition of new rule on already installed dedicated bearer.
   - Modification of existing rules (TFT change, MBR/GBR change).
   - Removal of existing rules.
   - Rule QCI change.

- Predefined Rules/GoRs
- Basic Support for ADC over dedicated bearers
- IDLE to ACTIVE mode transition (SAEGW, DDN) support for dedicated bearers.

6. Dedicated bearer deletion includes:
   - Deletion of dedicated bearer through MME/PCRF and **clear subscribers imsi** *imsi_id* **ebi** *ebi_id* CLI command.
7. During session recovery for Pure-P and Collapsed session at User Plane, charging data is recovered for User Plane.
8. MME and eNodeB Handovers (HO):
   - Pure-P Call Type:
     - MME and eNodeB HO with and without new policy (Create, Update, Delete, and any combination of Create, Update, and Delete) from Gx.
   - Collapsed Call Type:
     - MME and eNodeB HO with and without new policy (Create, Update, and Delete) from Gx.
9. S-GW Handovers (HO):
   - Pure-P to Pure-P HO:
     - Pure-P to Pure-P HO with and without dedicated bearer with new policy (Create, Update, Delete, and any combination of Create, Update, and Delete) from Gx.
     - Pure-P to Pure-P HO with bearer marked for deletion during HO.
   - Collapsed to Pure-P and Pure-P to Collapsed HO:
     - Collapsed to Pure-P and Pure-P to Collapsed HOs without dedicated bearer with new policy (Install new rule, modify default bearer QCI, update, or remove rule) from Gx.
     - Collapsed to Pure-P and Pure-P to Collapsed HOs with dedicated bearer with and without new policy from Gx.

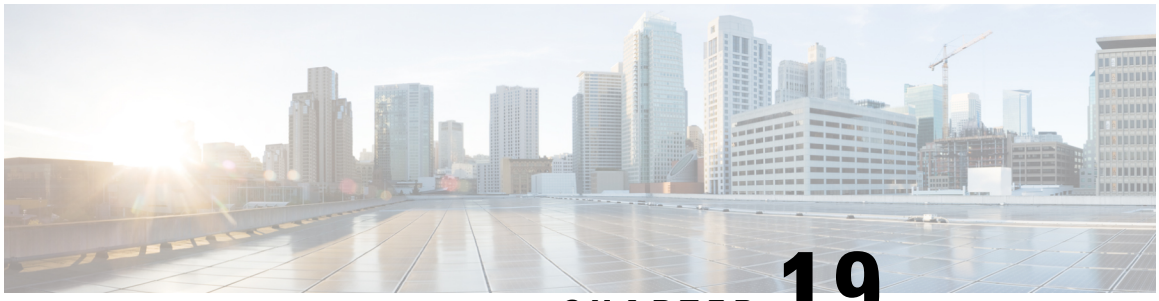
Limitations

In this release, the following functionality are not supported:

- Updation of dynamic rule precedence installed on default bearer.
- Time-based activation and deactivation of rules on default and dedicated bearer.
- Collision Handling is not yet supported.

  Collisions can happen between Control messages from PCRF and from Access side. Multiple procedures in a single PCRF initiated message (CCA-U/RAR) leads to uncontrolled collisions. For example, Creation of a Bearer along with Deletion of another Bearer in same RAR.

- Mid-session update and/or modification of ADC rules—whether change in configuration or PDN update over RAR, is not supported.
- MME and eNodeB Handovers (HO):
  - Pure-P Call Type:
    - Any failure handling or Collisions occurring during HO.
  - Collapsed Call Type:
    - MME and eNodeB HO with new policy (any combination of create, update, and delete together) from Gx.
    - Any failure handling or Collisions occurring during HO.
- S-GW Handovers (HO):
  - Pure-P to Pure-P HO:
    - Any failure handling or Collisions occurring during HO.
    - Dynamic rule QCI change installed on dedicated bearer such that its bearer EBI is not changed.
  - Collapsed to Pure-P and Pure-P to Collapsed HO:
    - Any failure handling or Collisions occurring during HO.



CHAPTER 19

Device ID in EDNS0 Records

- Revision History, on page 149
- Feature Description, on page 149
- How it Works, on page 150
- Configuring EDNS Format and Trigger Action, on page 153
- Monitoring and Troubleshooting, on page 155

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Table 6: Revision History

Revision Details	Release
The feature is available in 21.25 and later releases.	21.25
First introduced.	Pre 21.24

Feature Description

The Device ID in EDNS0 offers each enterprise with a customized domain blocking through Umbrella. To enable this functionality:

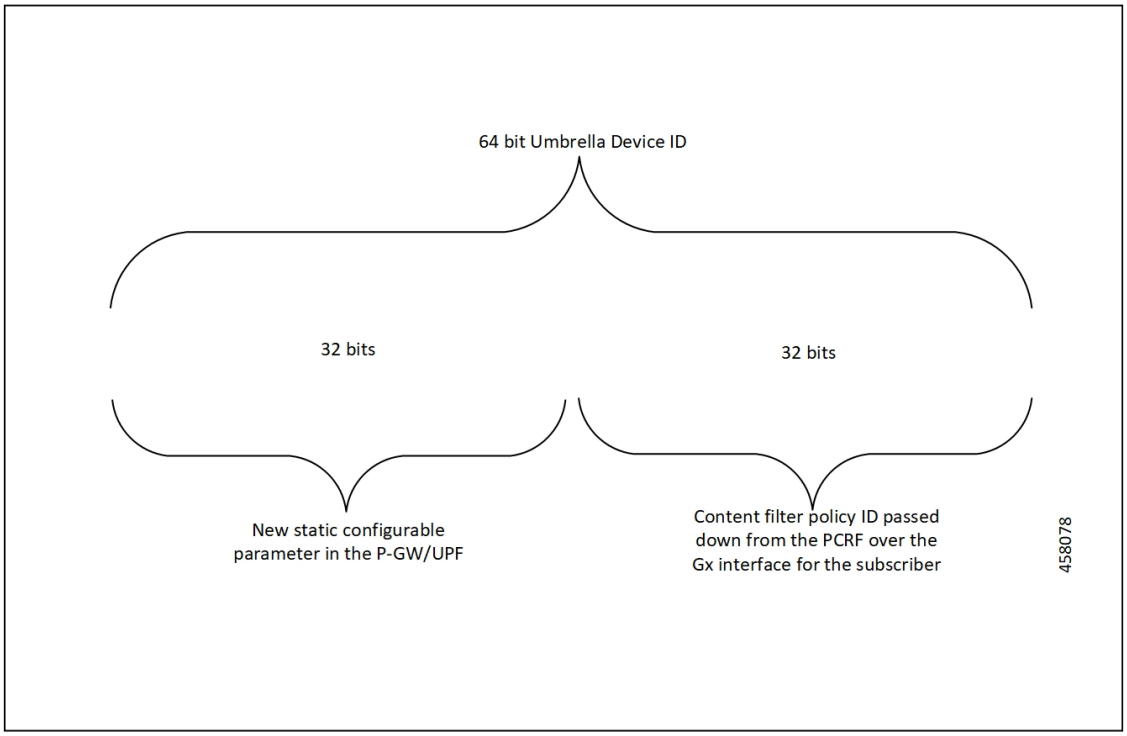
- The UP must reformat a subscriber DNS request into an EDNS0 request.
- The UP must include an Umbrella “Device ID” in the EDNS0 packet so that the Umbrella DNS resolver can use the Device ID to apply the domain filter associated/configured with the Device ID in the EDNS0 packet.

Presently, the Control Plane (CP) receives the domain filtering policy ID from PCRF/PCF. The CP passes the domain filtering policy ID to the User Plane (UP) in the Subscriber Parameters. The UP uses the domain filtering policy ID to apply domain filtering functionality to the subscriber.

How it Works

New CLIs are introduced to configure and trigger the EDNS0 functionality.

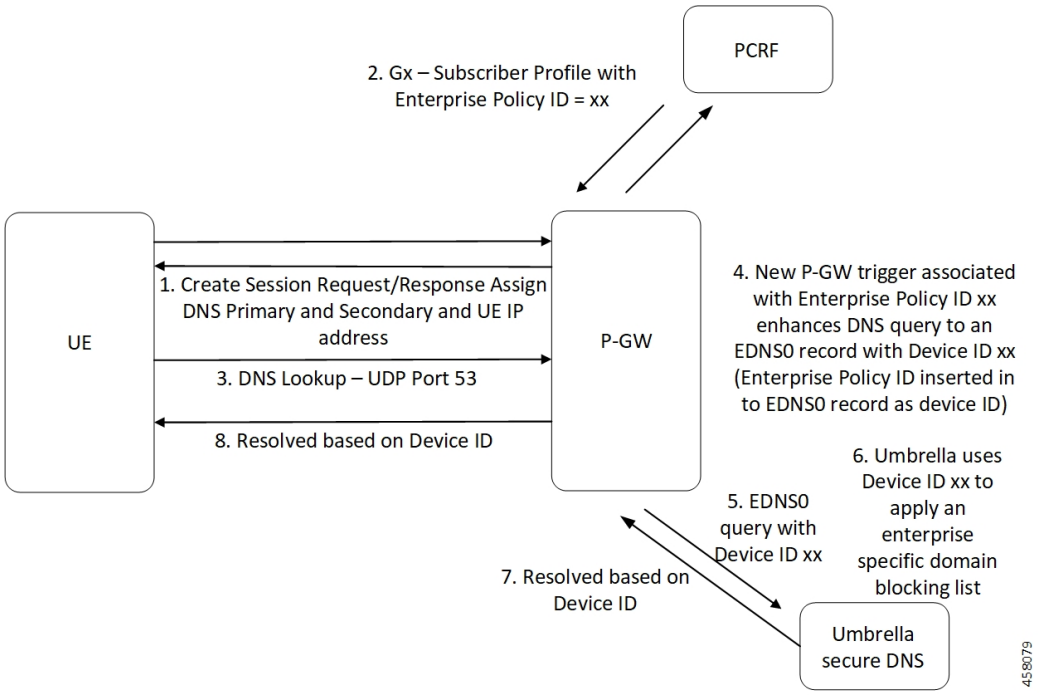
The EDNS0 packet receives the 64-bit device ID as OPT RR data. The first 32 bits of all device IDs is a fixed value configured in the UP. The last 32 bits of a subscriber device ID is the content filter ID value received from the PCRF/PCF. The UP concatenates the two 32-bit values to build a subscriber full 64-bit Device ID for populating in the subscriber EDNS0 queries. New CLI helps to configure the first 32 bit of static device-id value. If you don’t configure the 32-bit static prefix CLI, the outgoing packet shows the device-id = 32-bit CF PolicyID.



The Device ID number in the EDNS0 record allows the Umbrella DNS system to apply a custom set of domain filters for the EDNS0 queries.

Process Flow

The following process flow describes about the Content Filtering enhancement to insert Device ID in EDNS0 records:



EDNS0 Packet Format

The enterprise policy ID (CF_POLICY_ID) from PCRF helps to create the Device ID. The CP sends the device ID to the UP. Adding the Device ID to the DNS packet helps in creating the EDNS0 packet. The format of EDNS0 packets is specified by RFC2671. The following are few specifics:

- Following is the structure for the fixed part of an OPT RR:

```
Field Name   Field Type     Description
-------------------------------------------------
NAME         domain name    empty (root domain)
TYPE           u_int16_t          OPT
CLASS          u_int16_t          sender's UDP payload size
TTL            u_int32_t          extended RCODE and flags
RDLEN          u_int16_t          describes RDATA
RDATA          octet stream       {attribute, value} pairs
```

- Following is the variable part of an OPT RR encoded in its RDATA:

```
     +0 (MSB)                            +1 (LSB)
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 0: |                       OPTION-CODE                             |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 2: |                       OPTION-LENGTH                           |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 4: |                                                               |
   /                       OPTION-DATA                             /
   /                                                               /
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

  - OPTION-CODE: Assigned by IANA
  - OPTION-LENGTH: Size (in octets) of OPTION-DATA

• OPTION-DATA- Varies per OPTION-CODE

Example: If received policy-id from PCF/PCRF is “1234” and static prefix configured on UP is “5678”.

64-bits Device-ID will be “0000162e000004d2”.

- 0000162e -- 5678 (Decimal)
- 000004d2 -- 1234 (Decimal)

RDATA 69 42 00 0f 4f 70 65 6e 44 4e 53 00 00 16 2e 00 00 04 d2

- 6942 -- option-code
- 000f -- option-length
- 4f70656e444e53 -- OpenDNS (String)
- 0000162e -- 5678 (MSB)
- 000004d2 -- 1234 (LSB)

EDNS0 with IP Readdressing

The new CLI is configured within trigger action to readdress the DNS traffic to the Umbrella DNS. This CLI uses the existing readdress server list configuration from the ACS service. Readdressing of packets based on the destination IP address of the packets enables redirecting gateway traffic to configured server/port in the readdressed server list.

Behavior and Restrictions

Following are the behavior and restrictions applicable for this feature:

- Trigger Condition is evaluated at flow creation time. Any change in trigger condition in between the flow doesn’t affect the existing flow but affects the new flows.
- Any change to trigger action is applicable on the same flow.
- Neither CF nor EDNS is enforced when the CF Policy ID range is defined but Service-schema isn’t defined, or the Trigger condition pertaining to EDNS isn’t configured.
- If no CF Policy ID is received from Gx, range check isn’t performed, and content filtering works as defined in rule base.
- Cases where the ‘security-profile’ CLI isn’t associated with the ‘EDNS format’ CLI in Trigger Action, the device-id in the outgoing EDNS packet is sent with only 32-bit CF Policy ID.
- DNS queries with type other than A, AAAA, CNAME, NS, PTR, SRV, TXT, NULL aren’t to be EDNS converted.
- CF Policy ID change over Gx in between inflow aren’t applicable for the current flows. The current flows continue to insert the CF Policy ID present at the time of flow creation.

Limitation

Following are the limitations for this feature:

- The feature doesn't support the EDNS response packet reformat.
- The UP must be able to include the IMSI MSISDN tag value in the EDNS0 queries. This feature doesn't support the encrypted IMSI in EDNS0 packet. This feature also doesn't support the following configuration on the EDNS fields currently.

```
configure
   active-charging-service  service_name
      edns
         fields  fields_name
            tag default device-id
            tag 101 imsi encrypt
            tag 102 pgw-address
            end
```

Configuring EDNS Format and Trigger Action

Use the following configuration to enable DNS filtering:

```
configure
   active-charging-service service_name
      content-filtering range start_min_val to end_max_val
```

If the range parameter is set from 10 through 1000, any subscriber profile with a content filter policy ID from 10 through 1000 uses the standard content filtering functionality. Any subscriber profile with a content filter policy ID higher than 1000 or lower than 10 triggers the new EDNS0 functionality.

Use the following configuration to disable DNS filtering:

```
configure
   active-charging-service service_name
      no content-filtering range
```

When DNS filtering is disabled, the standard content filtering policies resume as configured or as received from the PCF.

Use the following configuration to configure the EDNS packet action and format under the active-charging service:

```
configure
   active-charging-service service_name
      trigger-condition trigger_condition_name
         external-content-filtering
           app-proto = dns
         end
```

NOTES:

- **external-content-filtering**: Enables EDNS0 feature. When this flag is true along with the range criteria, EDNS0 feature is enabled. By default, this flag is disabled.

- **app-proto = dns**: Avoids the IP readdressing of the non-DNS traffic. If this CLI is enabled with multiline-or cli, then all DNS traffic is EDNS encoded.

The following configuration leads the trigger action to define the EDNS format to be inserted in the EDNS packet:

```
configure
   active-charging-service service_name
      trigger-action trigger_action_name
         edns-format format_name
            security-profile profile_name
              flow action readdress server-list server_list_name [ hierarchy ]
 [ round-robin ] [ discard-on-failure ]
               end
```

NOTES:

- **trigger-action** *trigger_action_name*: Enables you to configure the flow action CLIs in the trigger action.
- **edns-format** *format_name*: Use the EDNS format when EDNS is applied.
- **security-profile** *profile_name*: Defines the security profile configuration in the EDNS to add mapping with the Device-ID.

Note Device ID in EDNS0 Records feature supports multiple security profiles.

- **flow action readdress server-list** *server_list_name* **[hierarchy] [round-robin] [discard-on-failure]**: Associates the EDNS with IP readdressing. Use IP readdressing to readdress the packets to the configured server IPs. This CLI in trigger action supports only server list configuration. It doesn't support single-server IP or port configuration like charging action.

Use the following configuration to insert the CF policy ID in the EDNS:

```
configure
   active-charging-service service_name
      edns
         fields fields_name
            tag { val { imsi | msisdn | cf-policy-id }}
            end
```

To configure the 32 MS bit, static value is provided at the EDNS level with the security profile.

```
security-profile security_profile cf-policy-id-static-prefix value
```

Use the following configuration to insert a new tag specifying the payload length:

```
tag default payload-length [ tcp | udp ] value
```

Value range: 576–4096

Sample Configuration

Following is the sample configuration for configuring the EDNS packets:

```
configure
active-charging service ACS
content-filtering range 10 to 100
ruledef dns-port
udp either-port = 53
tcp either-port = 53
multi-line-or all-lines
rule-application routing
#exit
 readdress-server-list re_adr_list_ta
  server 100.100.100.14
  server 2001::14
  server 100.100.100.15
  server 2001::15
#exit
rulebase starent
route priority 20 ruledef dns-port analyzer dns
#exit
edns
security-profile sec_profile cf-policy-id-static-prefix 123456
fields test_fields
tag 26946 cf-policy-id
#exit
format test_format
fields test_fields encode
#exit
#exit
trigger-action TA1
edns format test_format security-profile sec_profile
flow action readdress server-list re_adr_list_ta hierarchy
#exit
trigger-condition TC1
external-content-filtering
app-proto = dns
#exit
service-scheme SS1
trigger flow-create
priority 1 trigger-condition TC1 trigger-action TA1
#exit
subs-class SC1
rulebase = starent
multi-line-or all-lines
#exit
subscriber-base SB1
priority 1 subs-class SC1 bind service-scheme SS1
exit
end
```

Monitoring and Troubleshooting

Following are the show commands and outputs in support of enhance content filtering support to Insert device ID in EDNS0 records.

Show Commands and Outputs

Following are the show commands and outputs that are modified in support of the enhance content filtering support to Insert device ID in EDNS0 records.

- **show user-plane-service inline-services info**

```
CF Range: Enabled <<<<
  Start Value:  1 <<<<
  End Value:    1000 <<<
```

- **show user-plane-service statistics analyzer name dns**: output is modified to include the "EDNS Response Received" in both "EDNS Over UDP" and "EDNS Over TCP" sections.
- **show subscribers user-plane-only full callid**: output is modified to include the following parameters in the EDNS statistics per subscriber.
  - DNS-to-EDNS Uplink Pkts
  - DNS-to-EDNS Uplink Bytes
  - EDNS Response Received
- **show user-plane-service edns all**

```
Fields:
    Fields Name: fields_1
    tag 26946 cf-policy-id

    Fields Name: fields_2
    tag 2001 imsi
    tag 2002 msisdn
    tag 26946 cf-policy-id

    Format:
    Format Name: format_1
    fields fields_1 encode

    Format Name: format_2
    fields fields_2 encode

    Security-profile Name: high
    CF Prefix Policy ID: 1234
```

Use the following show commands to view the Trigger Action statistics:

- **show user-plane-service statistics trigger-action all**
- **show user-plane-service statistics trigger-action name** *trigger_action_name*
- **show user-plane-service trigger-condition all**

```
Trigger-Condition: TC1
       External-content-filtering : Enabled
       App-proto : dns
       Multi-line-OR All lines : Disabled
```

- **show user-plane-service trigger-action all**

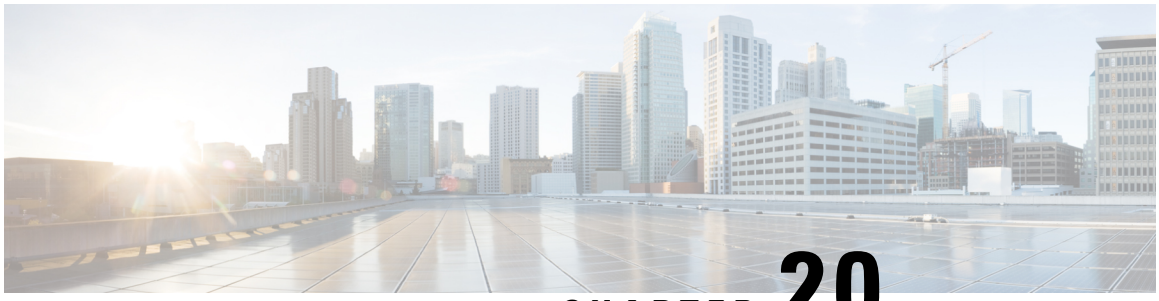
```
Trigger-Action: TA1
       HTTP Response Based TRM      : none
       HTTP Response Based Charging : none
       Throttle Suppress            : Disabled
       Flow Recovery                : Disabled
       Traffic Optimization         : Disabled
       Step Up GBR                  : Disabled
       Step Down GBR                : Disabled
       TCP Acceleration             : Disabled
       TCP Acceleration Threshold   : Disabled
       Service-Chain                : none
       UP-Service-Chain             : none
```

```
EDNS-Encode                  : Enabled
Flow-IP-Readdressing         : Enabled
```

Bulk Statistics

The following bulk statistics are available in support of the Device ID in EDNS0 Records feature:

SCHEMA: ECS	
Statistics	**Description**
ecs-dns-udp-edns-encode-succeed	The count of DNS to EDNS converted packets over UDP
ecs-dns-udp-edns-encode-failed	The count of failed DNS to EDNS conversions over UDP
ecs-dns-udp-edns-encode-response	The count of responses received for EDNS query over UDP
ecs-dns-tcp-edns-encode-succeed	The count of DNS to EDNS converted packets over TCP
ecs-dns-tcp-edns-encode-failed	The count of failed DNS to EDNS conversions over TCP
ecs-dns-tcp-edns-encode-response	The count of responses received for EDNS query over TCP



CHAPTER 20

DSCP Markings For Collapse Calls

- Feature Summary and Revision History, on page 159
- Feature Description, on page 159
- How It Works, on page 160
- Configuration, on page 160
- Monitoring and Troubleshooting, on page 161
- Show Commands Outputs, on page 161
- SMGR CP Changes, on page 161

Feature Summary and Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Currently QCI-based DSCP markings are applicable for Pure-S and Pure-P calls. The DSCP markings are based on QCI-QOS-Mapping associated with respective S-GW service or P-GW service. For collapse calls QCI-QOS-Mapping associated with PGW-service is applicable. This feature helps to apply the DSCP markings for collapse calls based on associated S-GW and P-GW services for uplink and downlink traffic. For uplink traffic DSCP markings associated with logical P-GW service is applicable. For downlink traffic DSCP markings associated with logical S-GW service is applicable. The DSCP markings are present in IP header of data traffic as a part of GTPU header and Inner IP. There's option to enabled or disable this functionality by CLI configuration. When you enable the feature, then only the new functionality is applicable otherwise existing functionality also works. By default, this feature is disabled so that there's no impact on customers who upgrades to this feature.

How It Works

Following are the steps that describes the DSCP markings for the collapse calls.

- In case of Collapse call:
  - for ACCESS side QCI-QOS mapping table associated with SGW-service is used.
  - For CORE side QCI-QOS mapping table associated with PGW-service is used.
- This is applicable once you enable the feature, otherwise QCI-QOS mapping table associated with PGW-service is used for both sides.
- APN associated QCI-QOS mapping table is preferred over the P-GW service QCI-QOS mapping table.
- APN-Profile associated QCI-QOS mapping table is preferred over SGW-Service QCI-QOS mapping table for ACCESS side DSCP markings.
- In case only P-GW service has QCI-QOS mapping table configuration then these DSCP markings is applicable on both ACCESS & CORE side for collapse call.
- In case only S-GW service has QCI-QOS mapping table configuration then these DSCP markings is applicable on ACCESS side for collapse call.
- There is a new configurable parameter inside the SAE-GW service which indicates whether the feature is enabled or disable.
- For Pure-P to Collapse HO and vice-versa, transport layer markings are updated in FAR as a part of Sx Modify request.
- Layer2 markings are also modified based on QCI-QOS mapping table picked for ACCESS & CORE side.
- DSCP markings continues to apply on existing bearers post session recovery.
- DSCP markings continues for the bearers on standby chassis once it switches to active mode.

Configuration

Configure the following command inside the SAE-GW service to enable/disable this feature.

```
configure
          context egress
             saegw_service saegw_service
             downlink-dscp-per-call-type enabled/disabled
             end
```

Note When you enable the feature, use the S-GW service QCI-QOS mapping DSCP markings for downlink, if call type collapses. By default, the downlink-DSCP-per-call-type is Disabled.

Monitoring and Troubleshooting

This section provides information on CLI commands that are available for monitoring and troubleshooting for DSCP markings for collapse calls.

Show Commands Outputs

This section provides information about show CLI commands that are available in support of DSCP markings for collapse calls.

show saegw-service all

This show command is to check if the feature is enabled or Disabled.

```
Service name : SAEGW11
Service-Id : 47
Context : EPC1
Status : STARTED
sgw-service : SGW11
pgw-service : PGW11
sx-service : SX11C
User Plane Tunnel GTPU Service : SAEGW11SXU
Newcall policy : n/a
downlink-dscp-per-call-type : enabled
CUPS Enabled : Yes
Service name : SAEGW21
Service-Id : 25
Context : EPC2
Status : STARTED
sgw-service : SGW21
pgw-service : PGW21
sx-service : SX21C
User Plane Tunnel GTPU Service : SAEGW21SXU
Newcall policy : n/a
downlink-dscp-per-call-type : disabled
CUPS Enabled : Yes
```

show sub user-plane-only callid <call_id> far full all

Use this User Plane CLIs to validate the Transport level marking options and inner packet markings for UPLINK/DOWNLINK FAR.

SMGR CP Changes

DSCP markings for Uplink/CORE and Downlink/ACCESS are present at bearer level inside sessmgr_sub_session_t → sessmgr_qci_tab_t.

User datagram DSCP markings are updated in IP header of inner packet, that is packet sent from UE to Internet and vice/versa.

Encaps header DSCP markings are updated in IP header of outer IP layer having GTPU-header (Outer header).

DSCP markings are sent from CP to UP inside FAR IE as follows:

- Transport Level Marking - The DSCP markings is configured in encaps header for ACCESS side and User-datagram on CORE side for collapse call.
- Transport Level Marking Options - Includes two options and are applicable only for outer header:
  - Copy-inner: Copy the inner packets markings to outer header
  - Copy-outer: Relay the DSCP markings for outer header

Inner Packet Marking - DSCP markings is configured in user datagram for ACCESS side. For CORE side it is N/A for collapse call.

Logic to fetch the DSCP marking changed for collapse call:

- Fetch the DSCP markings based on qci & qrp_pl for session from the associated SGW-Service for ACCESS/downlink side.
- Fetch the DSCP markings based on qci & qrp_pl for session from the associated PGW-Service for CORE/uplink side.
- For ACCESS/downlink side qci-qos mapping table associated with APN-profile takes preference over SGW-Service qci-qos-mapping table.
- For CORE/uplink side qci-qos mapping table associated with APN config takes preference over PGW-Service qci-qos-mapping table.
- In case SGW-service qci-qos mapping table is not configured, then PGW-service qci-qos-mapping table is applied on both ACCESS/CORE side.
- In case PGW-service qci-qos mapping table is not configured, then SGW-service qci-qos mapping tabled is applied on ACCESS side and no DSCP markings applied on CORE side.
- DSCP markings are updated on UP in create/update FAR sent as a part of SX Establishment/Modification request from CP to UP.
- Update the TLM, IPM & TLMO in case of HO from Pure-P to Collapse and vice-versa in Sx Modification request as a part of Update FAR IE.
- Update the layer2 markings in case of HO from Pure-P to Collapse and vice-versa in Sx Modification request as a part of Update FAR IE.

Following table depicts the various possible config combinations and outcome for DSCP markings to be applied on ACCESS and CORE side for COLLAPSE call:

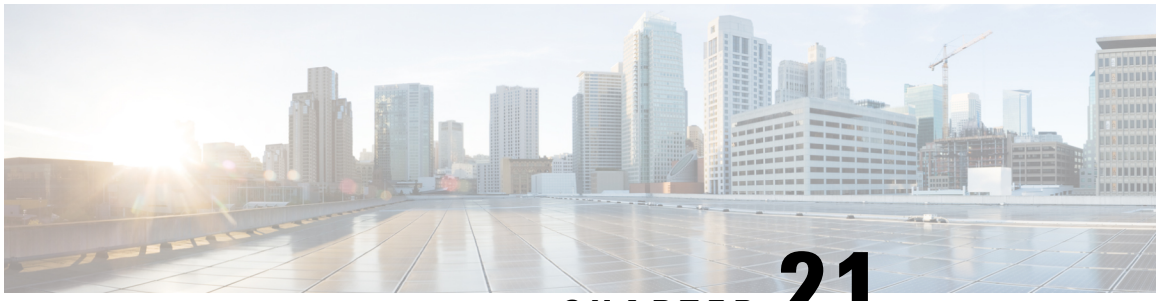
S. No.	Feature Enable/Disable	PGW Service QOS-QCI table configured(Q1)	SGW Service QOS-QCI table configured(Q2)	APN QOS-QCI table configured(Q3)	APN-Profile QOS-QCI table configured(Q4)	ACCESS/Downlink DSCP Markings for Collapse Call	CORE/Uplink DSCP Markings for Collapse Call
1	ENABLE	YES	YES	YES	YES	Q4 (APN-Profile)	Q3(APN)
2	ENABLE	YES	YES	YES	NO	Q2 (SGW-Service)	Q3(APN)
3	ENABLE	YES	YES	NO	YES	Q4 (APN-Profile)	Q1(PGW-service)
4	ENABLE	YES	YES	NO	NO	Q2 (SGW-Service)	Q1(PGW-service)

S. No.	Feature Enable/Disable	PGW Service QOS-QCI table configured(Q1)	SGW Service QOS-QCI table configured(Q2)	APN QOS-QCI table configured(Q3)	APN-Profile QOS-QCI table configured(Q4)	ACCESS/Downlink DSCP Markings for Collapse Call	CORE/Uplink DSCP Markings for Collapse Ca
5	ENABLE	YES	NO	YES	YES	Q4 (APN-Profile)	Q3(APN)
6	ENABLE	YES	NO	YES	NO	Q3(APN)	Q3(APN)
7	ENABLE	YES	NO	NO	YES	Q4 (APN-Profile)	Q1(PGW-service
8	ENABLE	YES	NO	NO	NO	Q1(PGW-service)	Q1(PGW-service
9	ENABLE	NO	YES	YES	YES	Q4 (APN-Profile)	Q3(APN)
10	ENABLE	NO	YES	YES	NO	Q2 (SGW-Service)	Q3(APN)
11	ENABLE	NO	YES	NO	YES	Q4 (APN-Profile)	N/A (NO DSCP
12	ENABLE	NO	YES	NO	NO	Q2 (SGW-Service)	N/A (NO DSCP
13	ENABLE	NO	NO	YES	YES	Q4 (APN-Profile)	Q3(APN)
14	ENABLE	NO	NO	YES	NO	Q3(APN)	Q3(APN)
15	ENABLE	NO	NO	NO	YES	Q4 (APN-Profile)	N/A (NO DSCP
16	ENABLE	NO	NO	NO	NO	N/A (NO DSCP)	N/A (NO DSCP
17	DISABLE	YES	YES	YES	YES	Q3(APN)	Q3(APN)
18	DISABLE	YES	YES	YES	NO	Q3(APN)	Q3(APN)
19	DISABLE	YES	YES	NO	YES	Q1(PGW-service)	Q1(PGW-service
20	DISABLE	YES	YES	NO	NO	Q1(PGW-service)	Q1(PGW-service
21	DISABLE	YES	NO	YES	YES	Q3(APN)	Q3(APN)
22	DISABLE	YES	NO	YES	NO	Q3(APN)	Q3(APN)
23	DISABLE	YES	NO	NO	YES	Q1(PGW-service)	Q1(PGW-service
24	DISABLE	YES	NO	NO	NO	Q1(PGW-service)	Q1(PGW-service
25	DISABLE	NO	YES	YES	YES	Q3(APN)	Q3(APN)
26	DISABLE	NO	YES	YES	NO	Q3(APN)	Q3(APN)
27	DISABLE	NO	YES	NO	YES	N/A (NO DSCP)	N/A (NO DSCP
28	DISABLE	NO	YES	NO	NO	N/A (NO DSCP)	N/A (NO DSCP
29	DISABLE	NO	NO	YES	YES	Q3(APN)	Q3(APN)
30	DISABLE	NO	NO	YES	NO	Q3(APN)	Q3(APN)
31	DISABLE	NO	NO	NO	YES	N/A (NO DSCP)	N/A (NO DSCP
32	DISABLE	NO	NO	NO	NO	N/A (NO DSCP)	N/A (NO DSCP

Statistics

Use the following User Plane CLI to show the number of TOS marked packets for U/L and D/L.

show sub user-plane-only full all



CHAPTER 21

Dynamic and ADC Charging Rule Names

- Revision History, on page 165
- Feature Description, on page 165

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

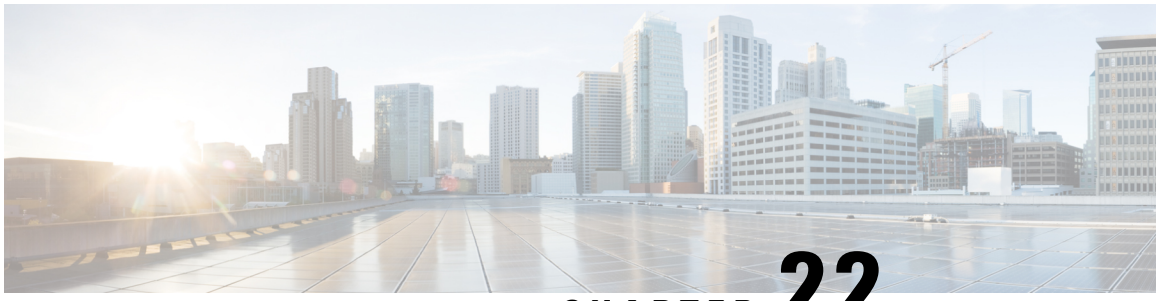
Feature Description

With this feature, the Operators can support Mobility Services Platform (MSP) functional use cases.

This feature covers the following requirements:

- Support of 64 rules is for:
  - Dynamic rules on default bearer with flow descriptions.
  - ADC rules with and without flow descriptions.
- Up to 174 PDRs, and it's corresponding FARs, URRs and QERs, are supported for static rules, predefined rules, dynamic rules, and ADC rules.
  - Up to 206 URRs are supported for static rules, predefined rules, dynamic rules, and ADC rules.
- All rules information is communicated to User Plane using Create PDR, Create URR, Create FAR and Create QER.
- All Sx messages supports the required number of PDR, URR, FAR, QER and Usage report, Query URR.
- Monitor protocol displays all Sx messages.

- Monitor subscriber displays all Sx messages.



CHAPTER 22

Dynamic APN and IP Pool Support

- Revision History, on page 167
- Feature Description, on page 167
- How It Works, on page 167
- Configuring Dynamic APN and IP Pool Support, on page 169

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The Dynamic APN and IP Pool Support feature enables the following functionality:

- Addition of an IP pool in an APN that previously had no IP pool configurations.
- Modification or removal of an existing IP pool configuration in an APN and adding a different one.
- Deletion or removal of existing IP pool configurations in an APN.

This feature supports dynamic configuration changes of the APN IP pools and groups and allocates the chunk to the User Plane (UP) without Sx reassociation.

How It Works

This section provides a brief of how the Dynamic APN and IP Pool Support feature works.

The Demux conveys the dynamically added APN and IP pool configuration to the VPN Manager. This information ensures the allocation of resources without Sx link breakage. The Control Plane (CP) then pushes the configuration to the User Plane via the Sx-Association Update message.

Dynamically Triggering APN IP Pool Addition Request

You can add APN and IP pools associated to new or existing APNs dynamically. During runtime, the new APNs and IP pools are added to the configuration. The configuration update occurs without causing any break in the Sx association between CP and UP,

The Dynamic APN and IP Pool Support feature also supports the following functionality:

- Addition of new IP pools or UP groups to existing APNs
- Addition of new APN to existing UP groups

This feature supports the following scenarios:

Operation on APN	Operation on IP Pool/Group	Operation on UP Group	Impact on existing calls
Addition of new APN	New pool or new group	New UP group (not registered)	No impact
Addition of new APN	New pool or new group	Existing UP group (already few UP registered)	No impact
Addition of new APN	Defaults pool	Existing UP group (already few UP registered)	No impact
Existing APN	Remove existing IP pool or group and add new IP pool or group	Existing UP group (already few UP registered)	No further allocation from the removed IP pool. No impact on calls.
Existing APN	Remove existing IP pool or group and add new IP pool or group	Default UP group	No further allocation from the removed IP pool. No impact on calls.
Existing APN	Remove existing IP pool or group from APN	Default UP group	No further allocation from the removed IP pool. No impact on calls.
Removing APN	Pass the new set of IP pools or groups to VPNMgr-C for each impacted UP	Existing UP group (already few UP registered)	All existing calls associated to that APN aren't affected but no new calls are connected.

Figure 8: Dynamic Addition of APN and IP Pools

After the newly added UP registration is successful, the VPN manager pushes the IP chunk information to the UP from the pool.

- The CP Sx-demux receives the trigger from the CLI for adding, modifying, or deleting a new APN or IP pools.
- The Sx-C demux on the CP determines the list of impacted UPs. It passes on the information for each impacted UP to the VPN manager at the CP using the Modify APN IP Pool Request.
- The VPN manager allocates the IP chunks and replies with a success or failure to the Sx-C demux.
- The new APN or IP pool is applied to the existing configuration. Use the "show config" CLI command to view the configuration.
- The addition of a new IP pool name or UP group to an existing APN does not affect the existing calls on that APN.
- Any IP pool (either IPv4 or IPv6) can be added to APN dynamically and can be modified (deleted and a new IP pool is added) in the same run. This change does not impact the existing calls in any way. The changed configuration applies only after the new calls to the APN are made.
- If any calls are running on a specific APN, any attempt to deleting that APN throws an error.
- Only APN that have no calls running can be deleted. The IP pools chunks associated to this APN are available for use to other APNs.

Passing of Allocated Chunks Information to the UP

- On receiving an Sx Association Update Request or Response message, the proprietary or custom IE pushes the IP chunk information to the UP.
- The S1-U demux on the UP passes on this information to the VPN manager on the UP
- The VPN manager receives the BGP routes, which it announces on a per chunk basis.

Limitations

The Dynamic APN and IP Pool Support feature has the following limitations:

- Any operation on IP pools and UP group associated to existing APN is not supported in this release.
- Multiple UPs won't be able to access the same IP pool as a part of this release.
- If a new UP_GROUP is added against an APN without configuring any new IP pool, then the calls start landing on the new UP_GROUP rather than the default one. This causes the calls to get aborted.

Configuring Dynamic APN and IP Pool Support

This section describes how to configure the Dynamic APN and IP Pool Support feature.

Follow this sequence of commands to add a new APN (addition of the IP pool is optional).

- Create a new IP pool. For more informattion, see the **ip address** *ip_ pool_name* CLI command in the *Command Reference Guide*.

  To add an IP pool to an IP pool group, use the **ip pool** *ip_pool_name* **static group-name** *ip_pool_group_name* CLI command. For more information, see the *Command Reference Guide*.

- Add a new APN and associate the new IP pool to this APN. For more informattion, see the **apn** *apn_name* CLI command in the *Command Reference Guide*.

  To add an IP Pool group to the APN, use the **ip address pool name** *ip_pool_group_name* CLI command. For more information, see the *Command Reference Guide*.
- Push the configuration to the UP.
- Update the IP pool information to the VPN manager.
- Run an attach call.

Updating the APN Configuration

Use the following command in Exec mode to update the VPN manager with the APN configuration changes.

To update all the configured APNs to the VPN manager:

```
update ip-pool apn all
end
```

To update a specific APN configuration to the VPN manager:

```
update ip-pool apn name apn_name
end
```

NOTES:

- This CLI command triggers the SX_ASSOCIATION_UPDATE towards the UP and transfers all the allocated IP pool chunks for the newly added IP pools.

Example

The following CLI command updates a specific APN configuration to the VPN manager:

```
update ip-pool apn name cisco.com
```

Verifying Dynamic APN and IP Pool Support

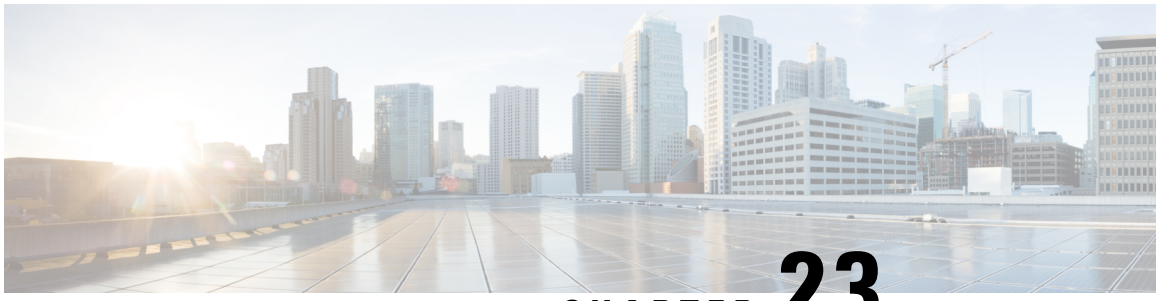
Use the following command to verify the Dynamic APN and IP Pool Support feature.

```
show config apn intershat
```

The following is a sample output of the show command:

```
config
  context ingress
    apn intershat
      pdp-type ipv4 ipv6
      bearer-control-mode mixed
      selection-mode subscribed sent-by-ms chosen-by-sgsn
      ims-auth-service ims-ggsn-auth
      ip access-group acl4-1 in
      ip access-group acl4-1 out
      ip context-name egress
      ip address pool name ipv4-test
      ipv6 access-group acl6-1 in
      ipv6 access-group acl6-1 out
      active-charging rulebase prepaid
```

```
    exit
  #exit
end
```



CHAPTER 23

Dynamic User Plane Selection

- Revision History, on page 173
- Feature Description, on page 173

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In a Multi-access Edge Computing (MEC) architecture, selecting an edge User Plane (UP) is key to providing low latency and maximum bandwidth efficiency. The location information of the user equipment (UE) is used to select an UP. For selecting an edge UP, two levels of granularity are considered, they are as follows:

- E-UTRAN Cell Global Identifier (ECGI) or Cell Global Identification (CGI) offers the lowest level of granularity.
- Tracking Area Identifier (TAI) or Routing Area Identity (RAI) or Service Area Identifier (SAI) offers the next level of granularity.
- TAI-SAI-RAI-ECGI offers fixed priority of TAI, SAI, RAI and ECGI in which ULI type is matched when more than one ULI type is received.

Architecture

To select a UP based on the location parameter of the upcoming session, a DNS Name Authority Pointer (NAPTR) query including TAI/RAI /SAI or ECGI/CGI is sent to the DNS Server. The DNS (NAPTR) Response contains a list of UP IPs. To select an UP from this list, a Load Control Information (LCI) and session count is applied to shortlist.

In this feature, the virtual APN selection is also enabled along with the dynamic UP selection. As a result, APN is selected based on the specified criteria. The selection criteria for the Virtual APN can also be based on location, for example, the Radio Admission Control (RAC) range.

Dynamic UP selection is done based on the **configure fqdn postfix** CLI command and the type of selected APN. If the type is ECGI or CGI, then a DNS Straightforward NAPTR (S-NAPTR) query is sent based on the cell ID. If the type is configured as tracking or routing area, then TAI or RAI or SAI is used for DNS (S-NAPTR) query.

To get the list of associated sx peers, UP group from the selected APN is used. The UP IPs in DNS (S-NAPTR) response is matched with the list of sx peers in the group. The peer that is either loaded least or have least sessions is selected from this list.

If ULI contains unsupported location data, Dynamic UP selection happens based on the RAI IE that comes outside ULI.

How it Works

This section describes the sequence of operation.

1. For P-GW, GGSN, or SAEGW, Fully Qualified Domain Names (FQDN) in UP, which contains **fqdn-postfix** and FQDN type (EGCI/CGI or TAI/RAI/SAI) are configured at APN level.
2. During an s6b interface protocol based authorization, the **fqdn-postfix** value in the authorization response is used (applicable for P-GW, GGSN, or SAEGW services only).
3. The DNS (S-NAPTR) query is sent to the DNS server.

Note The DNS (S-NAPTR) is generated based on the type(E-CGI | RAI-TAI-SAI | TAI-SAI-RAI-ECGI) configured in uplane fqdn at apn level for GGSN.

4. The response that is received from the DNS server is matched for service **x-3gpp-upf:x-sxb** for P-GW/GGSN/SAEGW(Collapsed) and **x-3gpp-upf:x-sxa** for S-GW.
5. The matching DNS (S-NAPTR) response is processed recursively for UP IPs.
6.
   - If enabled, the processed IPs are shortlisted for LCI based UP selection.
   - Or else, the processed IPs are shortlisted for session count based UP selection (with or without LCI).
7. If none of the UP IPs present in the response match with the associated sx peers, then, it leads to a session creation failure.
8. For S-GW dynamic UP selection, the DNS client context must be same as **sgw-service** context.
9. If there is a successful DNS response for S-GW dynamic UP selection, UPs are selected from the DNS dynamic list of UP addresses. If there is DNS failure (DNS response is empty without any UP address or DNS time-out), the UP selection falls back to the statically configured APN profile based user-plane-groups functionality.

Note

- Pure S-GW multi-PDNs work with independent DNS based UP selection.
- S-GW relocation use cases work with independent DNS based UP selection during a handover. If user-plane-group is configured under apn-profile, dynamic UP selection takes preference.
- After the DNS (NAPTR) query is sent, there are a few seconds of delay (equivalent to tx + rx) to receive the response.
- If the DNS server is not reachable, the session establishment might be delayed upto a maximum of 30 seconds before it uses the legacy method to select an UP.

The following sections describe various scenarios that are associated with the Dynamic UP Selection feature.

P-GW Dynamic UP Selection Having Virtual APN with Associated IP Pool

This section describes the sequence of operation for P-GW to dynamically select an UP having a virtual APN with an associated IP pool.

1. As part of the create session handling, PGW-C selects a virtual APN based on the TAC range.
2. The DNS (S-NAPTR) query is sent to the DNS server based on the configuration of the selected APN.
3. The response that is received from the DNS server is matched for service. The records with matching service fields are considered for selection.
4. The UP IPs that are part of a configured IP pool and present in the response are matched with the associated sx peers that are based on the UP group of the selected APN.
5. From the matching list, P-GW selects the UP that is the least loaded.

P-GW Dynamic UP Selection Having Virtual APN without Associated IP Pool

This section describes the sequence of operation for P-GW to dynamically select an UP having a virtual APN without an associated IP pool.

1. As part of the create session handling, PGW-C selects a virtual APN based on the TAC range.
2. The DNS (S-NAPTR) query is sent to the DNS server based on the configuration of the selected APN.
3. The response that is received from the DNS server is matched for service. The records with matching service fields are considered for selection.
4. The UP IPs that are part of any public IP pool and present in the response are matched with the associated sx peers that are based on the UP group of the selected APN.
5. From the matching list, P-GW selects the UP that is the least loaded.

S-GW Dynamic UP Selection for Successful DNS Response

This section describes the sequence of operation for S-GW to dynamically select an UP after receiving a successful response from the DNS server.

1. After an UE in a tracking area (or Cell ID) sends an attach request to S-GW with Dynamic ECGI, RAI-TAI-SAI | TAI-SAI-RAI-ECGI based UP selection feature is enabled and the DNS (S-NAPTR) query is sent to the DNS server.

2. S-GW receives the query response from the DNS server, which contains the list of UP IPs.
3. From the list of UP IPs, S-GW selects the UP that is the least loaded.

S-GW Dynamic UP Selection for DNS Response Time-out

This section describe the sequence of operation for S-GW to dynamically select an UP after the DNS server time-out or the server sends a negative response.

1. The S-GW sends the DNS (S-NAPTR) query to the DNS server.
2. If there is a DNS server time-out or the server sends a negative response after the DNS (S-NAPTR) query is sent to the DNS server, then S-GW selects an UP from the apn-profile UP group that are configured with static IPs.
3. From the list of UP IPs, S-GW selects the UP that is the least loaded.

Call Flows

This section includes the following call flows.

DNS Query Generation and Response Handling Call Flow

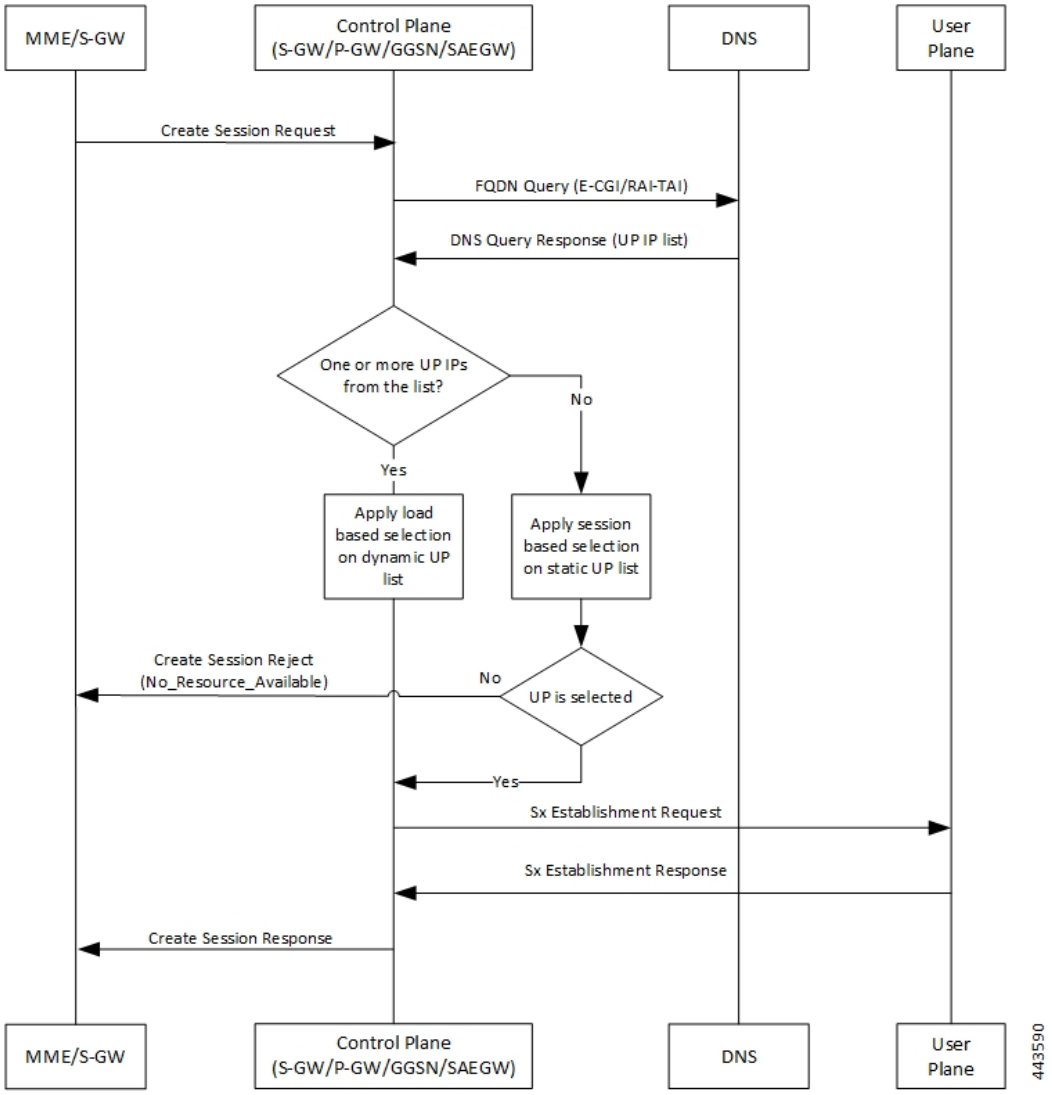


Table 7: DNS Query Generation and Response Handling Call Flow

Step	Description
1	MME or S-GW sends a Create Session Request message to the Control Plane (S-GW, P-GW, GGSN, or SAEGW).
2	Control Plane (CP) sends an FQDN query (E-CGI or TAI-RAI -SAI or TAI-SAI-RAI-ECGI) to the DNS server.
3	CP receives the response to the FQDN query with a list of UP IPs.

Step	Description
4	• If there are one or more UP IPs in the received list, CP applies LCI to the dynamic IP list to select an UP IP. • Or else, CP applies session count to the static IP list to select an UP IP.
5	• If an UP is selected, CP sends an Sx Establishment Request message is sent to UP (skip to step 6). • Or else, a Create Session Reject message is sent to MME or S-GW.
6	UP responds and sends an Sx Establishment Response message to CP.
7	CP sends a Create Session Response message to MME or S-GW.

DNS Query Timeout for Primary DNS Call Flow

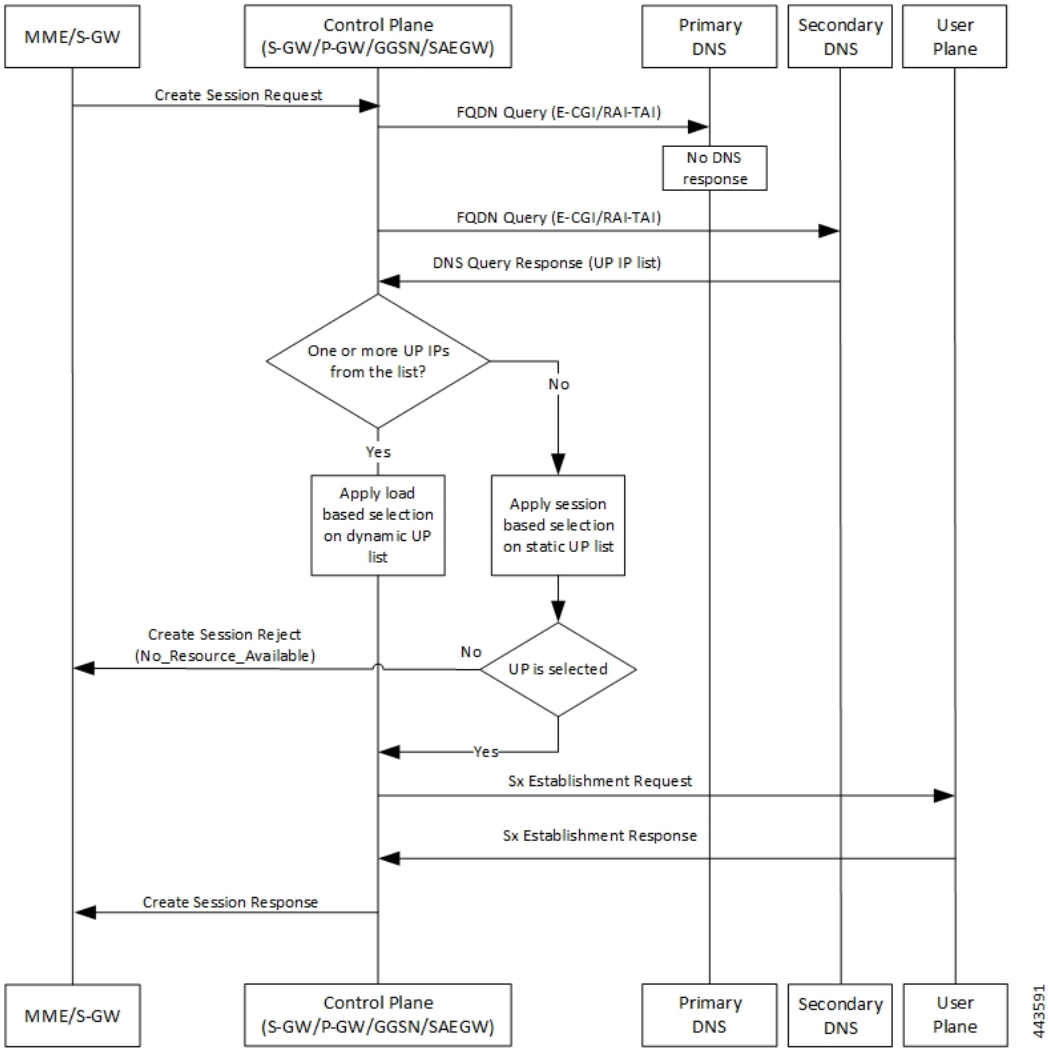


Table 8: DNS Query Timeout for Primary DNS Call Flow

Step	Description
1	MME or S-GW sends a Create Session Request message to the Control Plane (S-GW, P-GW, GGSN, or SAEGW).
2	Control Plane (CP) sends an FQDN query (E-CGI or TAI-RAI-SAI or TAI-SAI-RAI-ECGI) to the primary DNS server.
3	When there is no response to the query from the primary DNS server due to a time-out, CP retries to send the FQDN query to the secondary DNS server.
4	CP receives the response to the FQDN query from the secondary DNS server with a list of UP IPs.

Step	Description
5	• If there are one or more UP IPs in the received list, CP applies LCI to the dynamic IP list to select an UP IP. • Or else, CP applies session count to the static IP list to select an UP IP.
6	• If an UP is selected, CP sends an Sx Establishment Request message is sent to UP (skip to step 7). • Or else, a Create Session Reject message is sent to MME or S-GW.
7	UP responds and sends an Sx Establishment Response message to CP.
8	CP sends a Create Session Response message to MME or S-GW.

DNS Query Timeout for Primary and Secondary DNS Call Flow

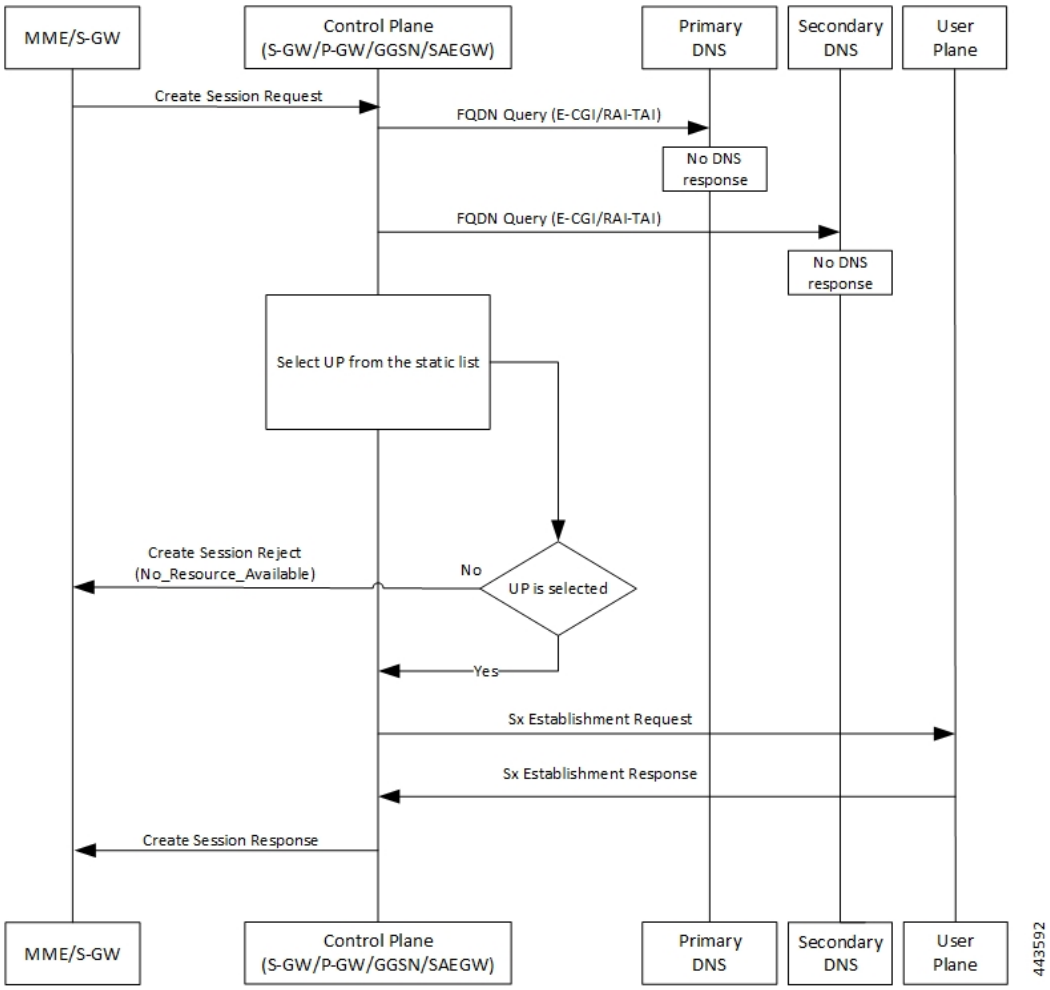


Table 9: DNS Query Timeout for Primary and Secondary DNS Call Flow

Step	Description
1	MME or S-GW sends a Create Session Request message to the Control Plane (S-GW, P-GW, GGSN, or SAEGW).
2	Control Plane (CP) sends an FQDN query (E-CGI or TAI-RAI-SAI or TAI-SAI-RAI-ECGI) to the primary DNS server.
3	When there is no response to the query from the primary DNS server due to a time-out, CP retries to send the FQDN query to the secondary DNS server.
4	When there is no response to the query from the secondary DNS server also, CP selects an UP IP from the list of static IPs.

Step	Description
5	• If an UP is selected, CP sends an Sx Establishment Request message is sent to UP (skip to step 6). • Or else, a Create Session Reject message is sent to MME or S-GW.
6	UP responds and sends an Sx Establishment Response message to CP.
7	CP sends a Create Session Response message to MME or S-GW.

Limitations

The Dynamic UP Selection feature has the following limitations:

- It's applicable for P-GW, S-GW, and SAEGW only.
- For SR and ICSR, no specific parameters are stored. If **smgr** is reset, the configured values are pushed again from **sessctrl**.
- Any changes to the DNS Server in not considered.
- Number of IPs handled for UP are limited to six, and this list of IPs can be a combination of IPv4 and IPv6 addresses.

Configuring the Dynamic User Plane Selection Feature

This section describes how to configure the Dynamic User Plane Selection feature.

Configuring CLI commands for P-GW or GGSN

```
configure
   context context_name
      apn apn_name
         user-plane-fqdn
         user-plane-fqdn fqdn_postfix_string type [ E-CGI | RAI-TAI -SAI |
TAI-SAI-RAI-ECGI ]
         end
```

NOTES:

- **user-plane-fqdn** Enables use of locally configured FQDN-postfix for dynamic UP selection (DNS-based).
- **E-CGI** Configures FQDN query type as E-CGI for UP selection
- **RAI-TAI-SAI** Configures FQDN query type as RAI-TAI-SAI for UP selection.
- **TAI-SAI-RAI-ECGI**: Configures FQDN query type as TAI-SAI-RAI-ECGI for UP selection.

Configuring CLI Commands for S-GW

```
configure
   context context_name
      sgw-service sgw-service_name
         user-plane-fqdn
         user-plane-fqdn fqdn_postfix_string type [ E-CGI | RAI-TAI -SAI |
TAI-SAI-RAI-ECGI ]
         end
```

NOTES:

- **user-plane-fqdn** Enables use of locally configured FQDN-postfix for dynamic UP selection (DNS based).
- **E-CGI** Configures FQDN query type as E-CGI for UP selection.
- **RAI-TAI-SAI** Configures FQDN query type as RAI-TAI-SAI for UP selection.
- **TAI-SAI-RAI-ECGI**: Configures FQDN query type as TAI-SAI-RAI-ECGI for UP selection.

Boxer Configurations

This section describes the following boxer configurations and restrictions.

1. DNS Client must be configured and associated with P-GW and GGSN service.
2. UP FQDN must be configured in APN.
3. IP addresses of the primary and secondary DNS servers must be configured in the ISP context.
4. UP FQDN must be configured in S-GW service for S-GW dynamic UP selection.

DNS Server Configurations

This section describes the following guidelines and restrictions to configure an external DNS server:

1. DNS must be configured for NAPTR to record for ECFI/CGI/TAI/RAI/SAI, as applicable.
2. NAPTR record must have service field as **"x-3gpp-upf:x-sxb"** for P-GW/SAEGW (Collapsed) and GGSN service and **"x-3gpp-upf:x-sxa"** for S-GW.
3. NAPTR record must have flags as **a** to indicate that the replacement string is FQDN for A or AAAA records.

The following CLI commands represent a sample DNS server configuration:

```
$ORIGIN 3gppnetwork.org.

$TTL 60              ; Put the Default
        TTL in seconds here (Its 1 day currently)

3gppnetwork.org.     IN SOA  nsbng.3gppnetwork.org. root.3gppnetwork.org.
        (

273 ; serial

7200 ; refresh (2 hours)

3600 ; retry (1 hour)

86400 ; expire (1 day)

43200 ; minimum (12 hours)
```

```
)

          NS
        nsbng.3gppnetwork.org.

ns AAAA 3001::41

;CUPS NAPTR Records Start From Here

;TAI NAPTR Records

tac-lb89.tac-hb67.tac.epc.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1  "a"
"x-3gpp-upf:x-sxb" ""
        uplane-address1-v4.3gppnetwork.org.

tac-lb89.tac-hb67.tac.epc.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1  "a"
"x-3gpp-upf:x-sxb" ""
        uplane-address1-v6.3gppnetwork.org.

tac-lb89.tac-hb67.tac.epc.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1  "a"
"x-3gpp-upf:x-sxa" ""
        uplane-address1-v4.3gppnetwork.org.

tac-lb89.tac-hb67.tac.epc.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1  "a"
"x-3gpp-upf:x-sxa" ""
        uplane-address1-v6.3gppnetwork.org.

;RAI NAPTR Records

rac34.lac-lb34.lac-hb12.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1  "a"   "x-3gpp-upf:x-sxb"
 ""
          uplane-address1-v4.3gppnetwork.org

.

rac34.lac-lb34.lac-hb12.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  2  "a"   "x-3gpp-upf:x-sxb"
 ""
        uplane-address1-v6.3gppnetwork.org.

;SAI NAPTR Records

sac1234.lac-lb34.lac-hb12.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1  'a'
'x-3gpp-upf:x-sxb' ''
           uplane-address1-v4.3gppnetwork.org.

sac1234.lac-lb34.lac-hb12.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  2  'a'
'x-3gpp-upf:x-sxb' ''
         uplane-address1-v6.3gppnetwork.org.

;ECGI NAPTR Records

eci-b167.eci-b245.eci-b323.eci-b401.eci.epc.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1
"a"   "x-3gpp-upf:x-sxb" ""
        uplane-address1-v4.3gppnetwork.org.

eci-b167.eci-b245.eci-b323.eci-b401.eci.epc.mnc365.mcc214.3gppnetwork.org. IN NAPTR 1  1
"a"   "x-3gpp-upf:x-sxb" ""
        uplane-address1-v6.3gppnetwork.org.

;CGI NAPTR Records

ci-lb34.ci-hb12.ci.lac-lb34.lac-hb12.lac.ggsn.mnc365.mcc214.3gppnetwork.org. IN NAPTR
          1  1
          "a"   "x-3gpp-upf:x-sxb" ""
        uplane-address1-v4.3gppnetwork.org.

ci-lb34.ci-hb12.ci.lac-lb34.lac-hb12.lac.ggsn.mnc365.mcc214.3gppnetwork.org. IN NAPTR
          1  1
```

```
          "a"   "x-3gpp-upf:x-sxb" ""s
        uplane-address1-v6.3gppnetwork.org.

;A Records

uplane-address1-v4   100     IN
          A    209.165.200.225

;uplane-address1-v4   100     IN      A
      209.165.200.225

uplane-address1-v4   100     IN
          A    209.165.200.225

;uplane-address2-v4   100     IN
          A    209.165.200.225

;AAAA Records

uplane-address1-v6   100     IN
          AAAA    1::1:111

uplane-address1-v6   100     IN
          AAAA    1111::1:111

;uplane-address2-v6   100     IN
          AAAA    1111::1:111
```

S6b Configuration (Optional)

This section describes guidelines to configure an external S6b to support custom attribute **aaa-uplane-fqdn** and to provide **fqdn_post_fix_string**.

```
AA-Answer
   apn-config
     uplane-fqdn
```

Interface

The following sections describe the format for the DNS query and response.

DNS (S-NAPTR) Query Format

This section describes the format for the DNS (S-NAPTR) query message.

Important SAI based FQDN is proprietary formatted and not as per the 3GPP Standard 23.003 19.4.2 Fully Qualified Domain Names.

Network Node	Query format
S-GW-C	**ECGI based** eci b1<ECI byte-1>.eci b2<ECI-byte-2>. Eci b3<ECI byte-3> .eci b4<ECI-byte-4>.eci.epc.mnc <MNC.mcc<MCC>.3gppnetwork.org **TAI based** tac lb<TAC low byte>.tac hb<TAC-high-byte> .tac.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org

Network Node	Query format
P-GW-C	**ECGI based** eci-b1<TAC-byte-1>.eci-b2 <ECI-byte-2.Eci-b3<TAC-byte-3> .eci-b4<ECI-byte-4>.eci.epc.mnc<MNC> .mcc<MCC>.3gppnetwork.org **TAI based** tac-lb<TAC-low-byte>.tac-hb<TAC-high-byte> .tac.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
GGSN-C	**CGI based** ci-lb<CI-low-byte>.ci-hb<CI-high-byte> .eci.lac-lb<LAC-low-byte>.lac-hb<LAC-high-byte> .lac.ggsn.mnc<MNC>.mcc<MCC>. 3gppnetwork.org **RAI based** rac<RAC>.lac-lb<LAC-low-byte> .lac-hb<LAC-high-byte>.lac.ggsn.mnc<MNC> .mcc<MCC>.3gppnetwork.org **SAI based** sac<SAC>.lac-lb<LAC-low-byte>. lac-hb<LAC-high-byte>.lac.ggsn mnc<MNC>.mcc<MCC>.3gppnetwork.org
SAEGW-C (Collapsed)	**ECGI based** eci-b1<TAC-byte-1>.eci-b2<ECI-byte-2> . Eci-b3<TAC-byte-3>.eci-b4<ECI-byte-4> .eci.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org **TAI based** tac-lb<TAC-low-byte> .tac-hb<TAC-high-byte>.tac.epc.mnc <MNC>.mcc<MCC>.3gppnetwork.org **SAI Based** sac<SAC>.lac lb<LAC low byte> .lac hb<LAC-high-byte>.lac.epc. mnc<MNC>.mcc<MCC>.3gppnetwork.org

DNS (S-NAPTR) Response Format

This section describes a sample format for DNS (S-NAPTR) response message.

```
Query ID              : 22290


Type                  : Response


Question              : NAPTR ?
          ci-lb34.ci-hb12.ci.lac-lb34.lac-hb12.lac.ggsn.mnc365.mcc214.3gppnetwork.org.


Answer                :


  Name                :
          ci-lb34.ci-hb12.ci.lac-lb34.lac-hb12.lac.ggsn.mnc365.mcc214.3gppnetwork.org.


  TTL                 : 60


  Type                : NAPTR


  Order               : 1


Preference : 1
  Flags               : a


  Service             : x-3gpp-upf:x-sxb


  Regexp              :


Replacement : uplane-address2.3gppnetwork.org.
  Name                :
        ci-lb34.ci-hb12.ci.lac-lb34.lac-hb12.lac.ggsn.mnc365.mcc214.3gppnetwork.org.
  TTL                 : 60
```

```
  Type                : NAPTR

  Order               : 1

  Preference          : 1

  Flags               : a

  Service             : x-3gpp-upf:x-sxb

  Regexp              :

Replacement : uplane-address1.3gppnetwork.org.
Query ID              : 44640



Type : Query
Question              : A?
        uplane-address2.3gppnetwork.org.
Query ID : 55480
Type                  : Query


Question              : A?
        uplane-address1.3gppnetwork.org.
Query ID : 55480
Type                  :
         Response


Question              : A?
         uplane-address1.3gppnetwork.org.


Answer                :



Name : uplane-address1.3gppnetwork.org.
  TTL                 : 100

  Type                : A
```

```
Address : 20.20.20.108
Query ID              : 44640


Type                  :
          Response


Question              : A?
          uplane-address2.3gppnetwork.org.


Answer                :


  Name
                       :
          uplane-address2.3gppnetwork.org.


  TTL                 : 100


  Type                : A


Address : 209.165.200.225
```

Show Commands

This section describes the available show commands for the Dynamic UP Selection feature.

show apn name *apn_name*

This show command can be used to check the following values:

- FQDN of APN
- Type of FQDN

show sgw-service *sgw-service_name*

This show command can be used to check the following values:

- FQDN of APN
- Type of FQDN

Statistics

This section provides information about the show CLI commands available to collect the statistics.

The following is a sample partial output for **show saegw-service statistics all** and **show saegw-service statistics name** *SAEGW21* CLI commands.

```
Dynamic Uplane Selection Statistics:
   Attempted              :        x
   Successful             :        x
   Failure                :        x
   Peer not Found         :        x
   Negative DNS response  :        x
   DNS timed out          :        x
   Unsolicited UP Selection Response:  x
   DNS Query Response post DNS timeout: x
```

The following is a sample partial output for **show saegw-service statistics all function sgw** CLI command.

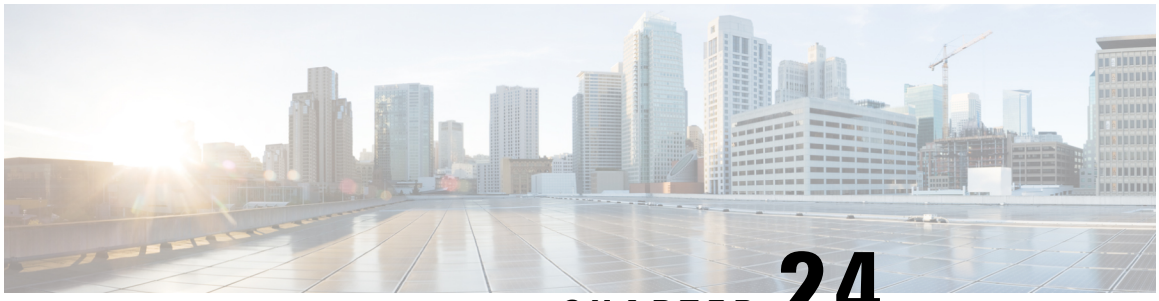
```
Dynamic Uplane Selection Statistics:
  Attempted:                              7
  Successful                                 4
  Failure:                                   3
    Mismatch DNS response:                1
    Negative DNS response:                1
    DNS timed out:                        1
    Unsolicited UP Selection Response:  1
  DNS Query Response post DNS timeout: 1
```

Bulk Statistics

saegw schema

Use this schema to collect the following bulk statistics pertaining to the Dynamic User Plane Selection feature:

- saegw-dyn-up-attempt
- saegw-dyn-up-attempt
- saegw-dyn-up-success
- saegw-dyn-up-success
- saegw-dyn-up-failure
- saegw-dyn-up-failure
- saegw-dyn-up-peer-not-found
- saegw-dyn-up-peer-not-found
- saegw-dyn-up-dns-timeout
- saegw-dyn-up-dns-timeout
- saegw-dyn-up-neg-resp
- saegw-dyn-up-neg-resp



CHAPTER 24

ECS Regular Expression Support

- Feature Summary and Revision History, on page 191
- Feature Description, on page 191
- How It Works, on page 192
- Configuring Regex Rule, on page 193
- Monitoring and Troubleshooting, on page 194

Feature Summary and Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

This feature provides the Enhanced Charging Support (ECS) for regular expression (regex) rule matching. The intent of the feature is to implement the regex engine in User Plane to enable RCM and PFD-based regex configuration/matching. The User Plane supports the following protocols as a part of regex engine rebuild and rule matching.

- HTTP
  - URL
  - URI
  - HOST
- WWW
  - URL
  - URI

- RTSP
  - URL
  - URI

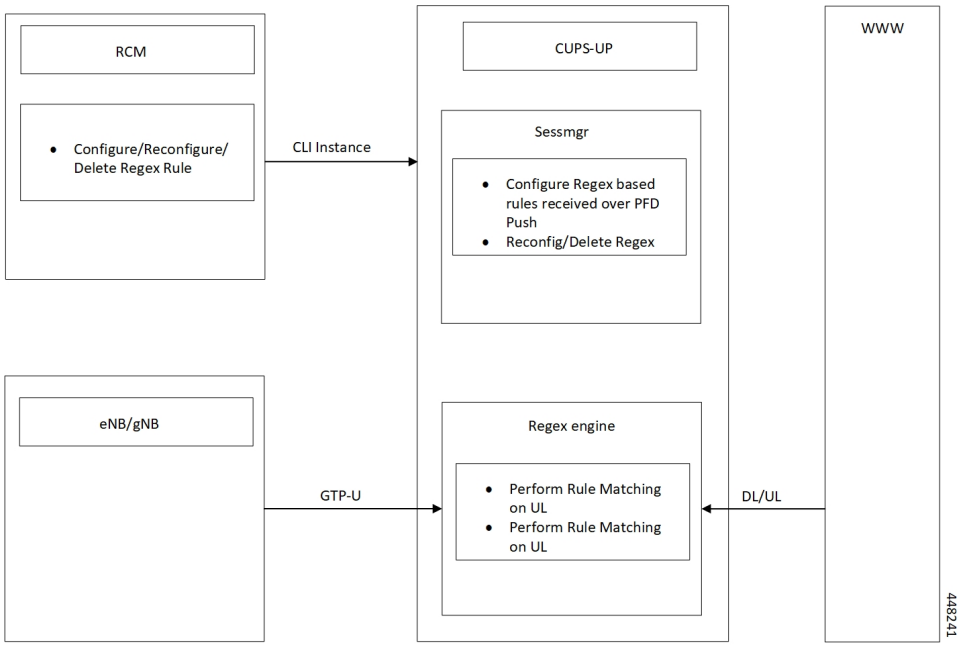
How It Works

The following table lists the special characters that you can use in regex rule expressions.

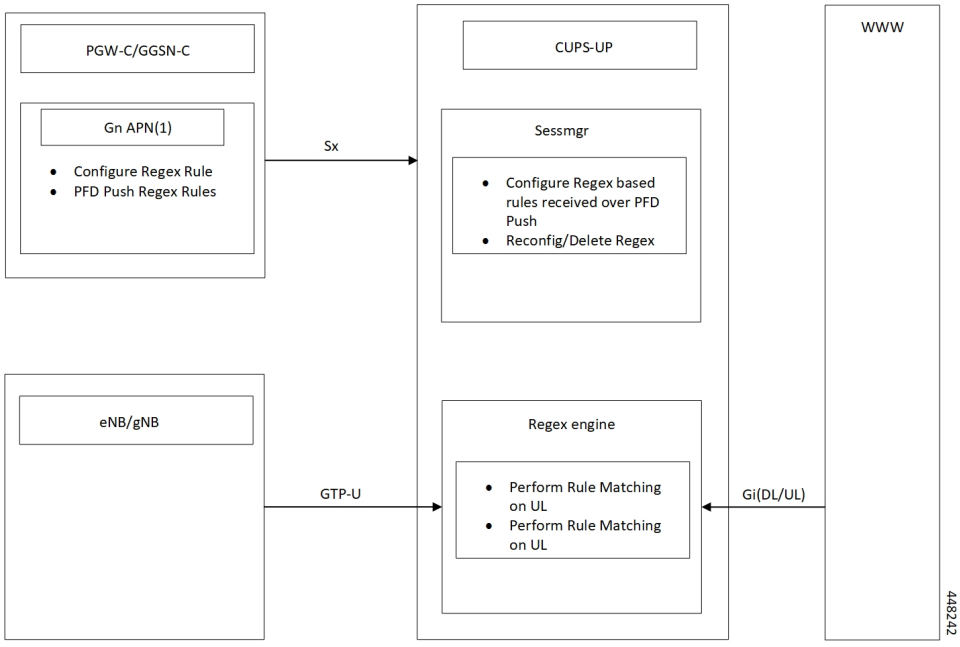
Convention	Description
*	Zero or more characters.
+	Zero or more repeated instances of the token preceding the +.
?	Zero or one character.
\character	Escaped character.
\?	Match on a question mark (\<ctrl-v>?)
\+	Match on a plus sign
*	Match on an asterisk
\a	Alert (ASCII 7)
\b	Backspace (ASCII 8)
\f	Form-feed (ASCII 12)
\n	New line (ASCII 10)
\r	Carriage return (ASCII 13)
\t	Tab (ASCII 9)
\v	Vertical tab (ASCII 11)
\0	Null (ASCII 0)
\\	Back slash
Bracketed range [0-9]	Matching any single character from the range.
A leading ^ in a range	No match in the range. All other characters represent themselves.
.\x##	Any ASCII character as specified in two-digit hex notation. For example, \x5A yields a ‘Z’.

Following are the two ways to configure the regex rule:

- Regex rule configuration via RCM:



• Regex rule configuration via PFD Push:



Configuring Regex Rule

Following are the two ways to configure the Regex Rule.

Configuring Regex Rule via RCM

Configure the regex rule via RCM through User Plane CLI instance or directly on User Plane via CLI.

```
configure
          require rcm-configmgr
          end
```

Configuring Regex Rule via PFD Push

Configure the regex rule on Control Plane through the User Plane via PFD push.

```
configure
        push config-to-up all
        end
```

Sample Configuration

Following are the sample configuration for configuring the Regex Rule.

```
configure
   active-charging service <service_name>
      ruledef <ruledef_name>
         http url regex <regex_url>
         rtsp uri regex <regex_uri>
         www url regex <regex_url>
         end
```

Note

- For RCM - Execute the regex rule configuration through the User Plane CLI instance.
- For PFD - Execute the regex rule configuration through the Control Plane and execute the PFD push.

Monitoring and Troubleshooting

This section provides information on CLI commands that are available for monitoring and troubleshooting for Regex support in User Plane.

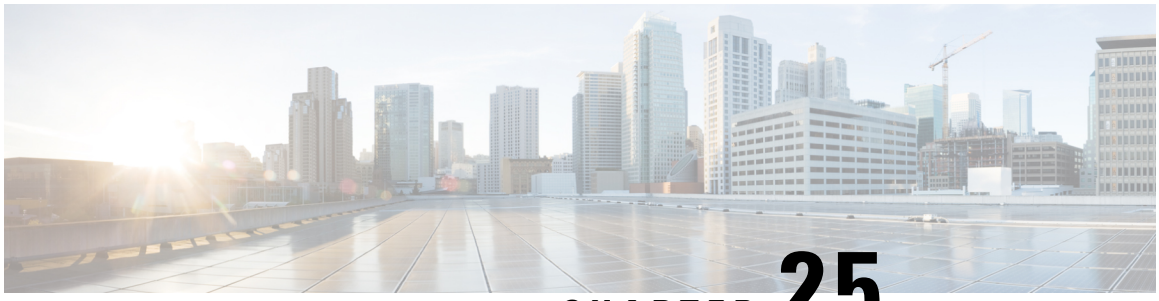
Show Commands and Outputs

This section provides information about show CLI commands that are available in support of Regex support in User Plane.

- **show user-plane-service regex status**: Use this command to display the engine status for SessMgr instance.
- **show user-plane-service regex statistics memory**: Use this command to display the memory stats for SessMgr instance.

- **show user-plane-service regex statistics memory summary**: Use this command to display the combined memory summary for the SessMgr.
- **show user-plane-service regex statistics ruledef**: Use this command to display the regex ruledef stats for the SessMgr.
- **show user-plane-service regex statistics ruledef summary**:

  Use this command to display the combined regex ruledef stats summary for the SessMgr.



CHAPTER 25

End Marker Packets

- Revision History, on page 197
- Feature Description, on page 197

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In case of eNodeB relocation during handover procedure without SGW-U change, the SGW-C indicates the SGW-U to switch the S1 path(s) by sending a Sx session modification request message with the new F-TEID-u of eNodeB. In addition, provides an indication to the SGW-U to send the End Marker Packet(s) on the old path. On receiving this indication, the SGW-U constructs End Marker Packet(s) and sends it for each S1 GTP-U tunnel toward the source eNodeB, after sending the last PDU on the old path.

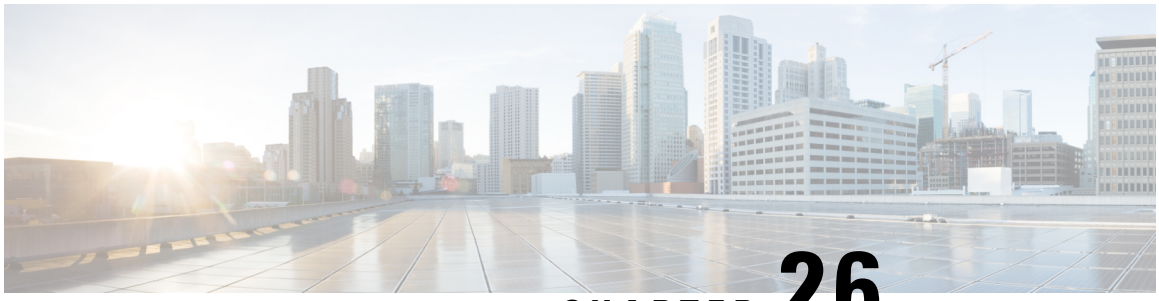
End Marker packet is sent per GTP-U TEID during above scenarios.

The Control Plane requests the User Plane to construct and send End Marker packets by sending a Session Modification Request including FAR(s) with the new downstream F-TEID, and with the SNDEM (Send End Marker Packets) flag set.

Information Element	P	Condition/Comment
PFCPSMReq-Flags	C	SNDEM (Send End Marker Packets): This IE shall be present if the CP function modifies the F-TEID of the downstream node in the Outer Header Creation IE and the CP function requests the UP function to construct and send GTP-U End Marker messages toward the old F-TEID of the downstream node.

Limitation

Handoffs in P-GW is not supported for sending End Marker. This behavior is similar to non-CUPS.



CHAPTER 26

Enterprise Onboarding in CUPS

- Feature Revision History, on page 199
- Feature Description, on page 199
- How it Works, on page 201
- Enterprise Onboarding in CUPS OAM Support, on page 218

Feature Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In CUPS architecture, User Planes (SAEGW-U) are grouped into a logical concept called User Plane Group (UP Group) and controlled by a Control Plane (CP) node. An APN is associated with a UP Group, and the UP for IP pool is selected based on least-used User Plane.

During configuration of new APNs and IP pools, the operator must decide on a UP Group to be used. The information required to decide the UP Group is not exposed by the system and the process is tedious and error prone. Also, the number of contexts, APNs, VRFs, and IP pools are reduced both on CP and UP in CUPS architecture as compared with ASR 5500. This also limits the addition of new APNs and IP Pools to the right context and UP Group.

The Intelligent Onboarding (IOB) tool automates the procedure of choosing the right UP Group and SGi context for the new APN to be added. The tool gathers current resources that are configured (number of UP Groups, UPs per group, existing contexts, APNs, and IP pools) in the CUPS system. It then determines if the system can absorb the new configuration and determines the UP Group that can support without breaching the system limits. In line with this, the new configuration is applied by the tool.

Operational Use Case

The Enterprise requires an operator to add, modify, and/or delete a user with information based on APN and IP pools. The tool generates and applies the required configuration to add, modify, or delete an APN in the CUPS environment.

The following operations can be performed:

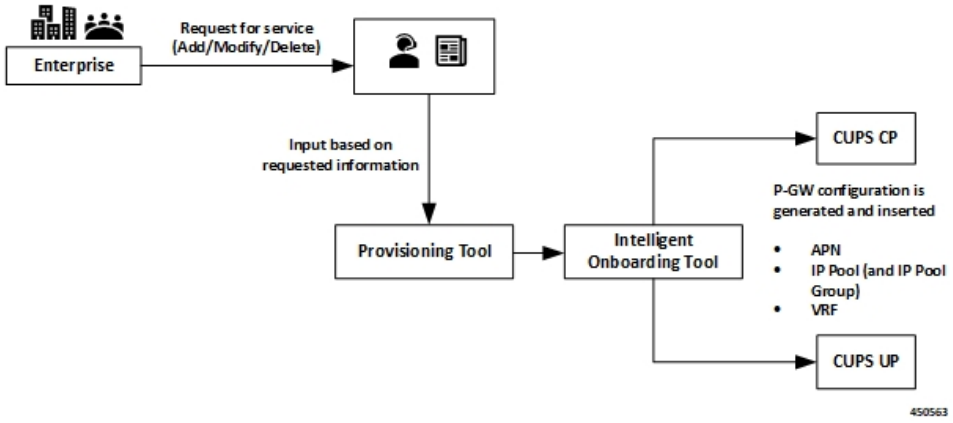
- Enterprise Addition: A new APN is added with required number of IPv4/IPv6 pools.
- Enterprise Modification: IP pools can be added/deleted for an existing APN.
- Enterprise Deletion: An APN will be deleted.

In 21.20.13 and later releases, the IOB tool also supports the onboarding of one or more virtual APNs in one operation. As part of this operation, one or more existing APNs can be modified to reference these new virtual APNs. Similarly, the tool also supports deleting the set of virtual APNs onboarded together and simultaneously removing existing references to those APNs from other APNs.

Architecture

On ASR 5500, Enterprise addition consists of adding a new APN. For CUPS, along with the APN configuration, we must include the correct UP Group and SGi context configuration.

The IOB tool takes inputs from the Provisioning tool, chooses the best suited UP Group and SGi context for the APN, and configures the CP and UP. The IOB tool also allows modification of the APN configuration (adding/deleting the IP pools) and Deletion of an APN.



If onboarding multiple APNs, the APN configuration section must list:

- The configuration for all onboarding APNs and
- APNs referring to them (in case of virtual APNs)

In the preceding scenario, all the APNs get onboarded onto the same UP group and SGi context.

Installation

The IOB tool is shipped as Linux executable. All dependencies, like Pexpect and connection management library, are packaged into the standalone .exe file.

The tool is shipped with StarOS images and signed with the same keys that are used for StarOS VPC-SI image.

The executable tool requires the following environment:

- RedHat Enterprise Linux 7.6 (or CentOS equivalent) 64-bit installation
- OpenSSL version 1.0.2.k-fips
- The following shared libraries are installed under /lib64 (these are typically present in a standard RHEL or CentOS installation):
  - libdl.so.2
  - libz.so.1
  - libc.so.6
  - ld-linux-x86-64.so.2
- Read, write, execute permissions for */tmp* directory. While executing, the tool creates a temporary directory under */tmp*, extracts sections of the executable to this temporary directory and executes the sections.
- Sufficient disk space for the tool and the log files (current usage is approximately 10 MB)
- IP connectivity to CPs and UPs on which onboarding is to be done. Password-based SSH is used for connections.

How it Works

The IOB tool is a standalone application that leverages StarOS CLIs to collect the system level resources, read the configurations, check the errors, SRP information, and so on. The input parameters to the IOB tool include addressing and login credentials for CPs and UPs, details of the operation (add/modify/delete), and the specific configuration to be applied. Since the contexts to apply the configuration to may not be known beforehand, the input configuration specifies a dummy context as a placeholder. The IOB tool substitutes that dummy context with the specific context that is chosen prior to applying the configuration.

Also, as part of Enterprise Onboarding solution, a new CLI command is introduced, and an existing CLI command is modified. For details, see *Enterprise Onboarding in CUPS OA&M Support* section.

The IOB tool goes through the following steps:

- **Pre-processing**: This is performed to ensure that the system is in stable state to proceed with the onboarding operation. On successful validation, the IOB tool collects the current resource usage information from the system.
- **Context and UP Group Selection**: The IOB tool applies the onboarding algorithm to select a context and UP Group to onboard the APNs.
- **Configuration**: Based on the operation to be performed, an algorithm is applied using the data collected in the Pre-processing step. The configuration is then applied on CP and UP. For any failure scenario, the IOB tool attempts to roll back to the previous configuration.

- **Post-processing**: Post configuration checks are performed to validate the system for any errors. For any failure scenario, the IOB tool attempts to roll back to the previous configuration.
- **Logging**: The entire operation is logged. The logging mechanism captures the output of the operation, history of the operation, Warnings/Error messages, and any other information that helps in debugging.

Pre-Processing

Pre-processing step helps in understanding the status of the CUPS system where the onboarding operation is being performed. In the pre-processing stage, following checks are performed irrespective of the operation:

- Verify if all CP and UP management IPs are reachable:
  - Ping Active/Standby management IPs of all the CPs.
  - Ping Active/Standby management IPs of all the UPs.
- Collect the resources information (APN, IP Pools, VRF, and Context) based on the output of:
  - **show ip user-plane verbose**
  - **show cups-resources session summary**
- **Add Operation**:
  - On Control Plane node, following checks are performed:
    - Verifies that the VRF, APN, and IP pool to be onboarded is not configured in the system. If onboarding one or more virtual APNs, then the APNs that refer to these virtual APNs must be already present on the system. The tool uses the presence of the following configuration in the APN to distinguish these APNs.

      ```
      virtual-apn gcdr apn-name-to-be-included Gn
      ```

      So, given an input configuration with one or more APNs, then any APN that is already present in the system must include the preceding configuration. Otherwise, the tool assumes that the APN isn't present and hence fails the preaudit step.
    - Verifies that there is no configuration difference between Active/Standby CPs using **show srp info**.
  - After context and UP Group selection, on User Plane node, the following pre-processing checks are performed on all the UPs of the selected UP Group:
    - Verifies that the VRF to be onboarded doesn't exist in the system. If it exists, then the pre-processing fails and onboarding is aborted.
    - Verifies that there is no configuration difference between Active/Standby UPs using **show srp info**.
    - Verifies if SGi context is mapped in the UP Groups.
- **Modify Operation**:
  - On Control Plane node, following checks are performed:
    - Verifies that the VRF to be modified exists in the system.

- Verifies that the APN to be modified exists in the system.
- Verifies that the IP pools, deleted as part of modify operation, exists in the system. Any IP pool that is added as part of modify operation, doesn't exist in the system.
- Verifies that there is no configuration difference between Active/Standby CPs using **show srp info**.

- **Delete Operation**:
  - On Control Plane node, following checks are performed:
    - Verifies that the VRF to be deleted exists in the system.
    - Verifies that the APN(s) to be deleted exists in the system.
    - Verifies that there is no configuration difference between Active/Standby CPs using **show srp info**.
  - On User Plane node, following checks are performed:
    - Verifies that the VRF to be deleted exists in the system.
    - Verifies that there is no configuration difference between Active/Standby UPs using **show srp info**.

CP and UP Configuration

On successful pre-processing, the tool performs the Add/Modify/Delete operation as per the input and applies the configuration on CP and UP. For ICSR setups, the configuration is applied on both Active and Standby CP and UPs.

- Add operation: The algorithm chooses the right SGi context and UP group for the APN to be added.
  - On Control Plane node, following steps are performed:
    - The chosen SGi context and the UP Group are added to the APN configuration, which goes as input to the tool. In case of onboarding virtual APNs, only the onboard virtual APNs get updated with UP Group and IP context. The APNs that refer to them (which is already present in the system) just gets updated with any **virtual-apn preference ..** configuration that is present in the input file.
    - The updated configuration is then applied to the CP node.
  - On User Plane node:
    - The IOB tool replaces the dummy SGi context with chosen context, and applies the resulting configuration to all the UPs in the chosen UP Group.
    - Applies VRF configurations to all the UPs in the UP Group.
  - For any failure scenario, the IOB tool attempts to roll back to the previous configuration.
- Modify Operation: Configuration is modified to add or delete the IP pools.

- On Control Plane node:
  - For the given APN configuration, IP pool configuration is modified to add/delete the IP pools. If any IP pools are deleted, then prior to deletion, the tool:
    - Busyouts the pool.
    - Clears existing subscribers for that pool per pace-out interval. The pace-out interval is calculated based on the size of the pool.

      For IPv6 pools, the formula is:

      Pace-out interval = (2^(64 - pool size) * 2 - 2) / 500

      So, a /48 pool will get a pace-out interval of (2^(64 - 48) * 2 - 2) / 500 = (2^16 * 2 - 2) / 500 = 131070 / 500 = 262 seconds

      For IPv4 pools, the formula is:

      Pace-out interval = (2^(32 - pool size) * 2 - 2) / 500

      So, a /21 pool will get a pace-out interval of (2^(32 - 21) * 2 - 2) / 500 = (2^11 * 2 - 2) / 500 = 4094 / 500 = 8 seconds

  - For any failure scenario, the IOB tool attempts to roll back to the previous configuration.

- Delete Operation: Deletes the APN.
  - On Control Plane node:
    - IP pools and VRFs, associated with the APN, are deleted.

      Prior to deleting any APN, the IOB tool verifies if any user is attached to the given APN. If any user exists, it exits from the tool and displays an error message "Please clear the subscribers then run the DELETE_ENTERPRISE else it will delete the APN".

    - APN configuration is deleted.
    - Deleting the virtual APNs removes only the virtual APNs and references to the virtual APNs. The APNs that refer to them are expected to remain in the system. Otherwise, post audit will fail.

  - On User Plane node, VRF configurations are deleted.

    The IOB tool doesn't rollback to the previous configuration on a failure. It, however, tries to delete as much of the relevant configuration as possible to minimize the amount of manual clean-up required.

Post-Processing

After the configurations are pushed to CP and UP, checks are performed to validate configuration changes.

- Add Operation:
  - On Control Plane node, following checks are performed:

- Verifies configured VRF with **show ip vrf** *vrf_name*: To verify if the VRF configuration is applied in the CUPS system.
- Verifies that the chosen context is shown with **show configuration apn** *apn_name*: To verify if the context has been associated with the APN that is added. This verification takes place for each APN that is onboard. If there are virtual APNs onboard, then this verification takes place only for each virtual APN.
- Verifies that the chosen UP Group is shown under **show configuration apn** *apn_name*: To verify if the UP Group has been associated with the APN that is added.
- If there are virtual APNs onboard, then the tool verifies that all the references to the virtual APNs from other APNs as per the input configuration (**virtual-apn preference** <*preference*> **apn** <*virtual apn*> and so on) are present and correct.
- Saves configuration using **save configuration** *file_path* / *file_name*: After successful addition of new enterprise, checks if the respective configuration files are stored in the given path as mentioned in "CUPSinfo.txt" file.
- Synchronize configuration on CPs with **filesystem synchronize**: After successful addition of new enterprise, verifies the file synchronization.
- Verifies that there is no configuration difference between CPs using **show srp info**: SRP validation in ICSR setup: After successful addition of new enterprise, the IOB tool checks for SRP validation with "Primary" and "secondary" status, "Last Peer Configuration Error", "Connection State", along with "Number of Sessmgrs".

- On User Plane node, following checks are performed:
  - Verifies configured VRF with **show ip vrf** *vrf_name*: To verify the VRF configuration applied in the CUPS system.
  - Verifies Route Distinguisher using **show ip vrf** *vrf_name*: To verify the Route Distinguisher configuration applied in the CUPS system.
  - Save configuration using **save configuration** *file_path* / *file_name*.
  - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup.
- For any failure scenario, the IOB tool attempts to roll back to the previous configuration.

- Modify Operation:
  - On Control Plane node, following checks are performed:
    - Verifies that the modified changes are applied to the CUPS system.
    - Verifies that the changes to IP pool are reflected in the system
    - Saves configuration using **save configuration** *file_path* / *file_name*.
    - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup
  - For any failure scenario, the IOB tool attempts to roll back to the previous configuration.

- Delete Operation:
  - On Control Plane node, following checks are performed:
    - This verification takes place for each APN after the delete operation. The delete operation for the virtual APN removes only the references to the virtual APN and retains the APNs that refer to them. Removal of the latter APNs causes the post processing to fail.
    - Verifies if the VRF configuration is deleted from the CUPS system.
    - Saves configuration using **save configuration** *file_path* / *file_name.*
    - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup
  - On User Plane node, following checks are performed:
    - Verifies if the VRF configuration is deleted from the CUPS system.
    - Saves configuration using **save configuration** *file_path* / *file_name.*
    - Invoking SRP validation using **srp validate-configuration**: Verifies that there is no configuration difference between UPs using **show srp info**: SRP validation in ICSR setup.

Add Operation

The Add operation configures a new APN for the enterprise customer. The tool also supports onboarding multiple APNs in one operation, provided they share the same SGi context and VRF configuration. In this case, the onboarding APNs may or may not share the IP pool information (supports both the conditions). All the onboarding APNs map to the same SGi context and UP group in the preceding scenario. The algorithm chooses the right SGi context and UP Group, and maps them to the APN by taking the system parameters into consideration.

Algorithm Logic:

- Check for System Limits (done against CP limits mentioned in System Limits, on page 216). In case of onboarding virtual APNs, the tool only considers the virtual APNs as new APNs for APN limit calculation. The APNs referring to the virtual APNs are already present in the system and are hence already included in the current count of APNs in the system.
- Rank UP Groups based on number of APNs configured with low numbers on top.
- Sort the SGi contexts based on the number of VRFs configured in ascending order.
- Exclude VIP UP Groups and Contexts from the list.
- Pick a UP Group from top of the list (least-used):
  - Get a Context that is mapped to UP Group (if no Contexts are mapped, pick from the sorted list).
  - Check the number of VRFs, IPv4, IPv6 pools, and total pool size.
  - Choose the Context if checks fall within thresholds; else, repeat for next Context.
  - Pick suitable Context within limits; if none found, exit algorithm.
  - For this UP Group, iterate through UPs and check total IP pool limits.

    - If successful, choose the UP Group and Context.

- Iterate through all UP Groups.
- At each step, while checking against the thresholds, print error messages.
- Prepare the configuration with chosen Context and UP Group and apply.

Modify Operation

Modify operation allows the onboarded Enterprise customer to increase/decrease the subscribers by Adding more IP pools or deleting the existing IP pools.

Delete Operation

Delete operation removes a previously onboarded Enterprise. During this operation, the IOB tool cleans up the IP pools, VRFs, and APNs that are used for the Enterprise.

To delete an Enterprise, the following procedure must be followed as there may be active subscribers on the system:

- Busyout the IP Pools: This is performed to block the new subscribers. Invoke IOB tool and perform the Busyout operation using the MODIFY operation.
- Clear Subscribers: The Provisioning tool clears the active subscribers.
- Delete the Enterprise: Invoke the IOB tool and perform the enterprise removal using the DELETE operation.

Password Encryption

The IOB tool expects passwords in the "CUPSInfo.txt" input file to be RSA encrypted and converted to base64 format. Encryption is done using OpenSSL (currently, version 1.0.2.k is supported) commands and RSA public key. The IOB tool must be provided the path to the corresponding RSA private key so that it can decrypt the passwords. The decrypted passwords are stored only in the IOB tool's RAM. The detailed steps for encryption and decryption are described below:

1. Verify that the OpenSSL with the correct version is installed on the target machine:
    - "openssl version" should indicate that the version is 1.0.2.k-fips.
2. Generate RSA private and public key pair:
    a. RSA private key:

    ```
    openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:4096
    ```

    Where:

    - "private_key.pem" represents the generated private key file in PEM format. This is used for decryption and has to be stored securely.

- 4096 is the key length in bits. Either 2048 or 4096 can be used. Multiple passwords may need to be encrypted and so, 4096 is recommended. Generally, the larger the key size, the larger the size of data it can encrypt. However, it also takes longer to encrypt/decrypt.

b. RSA public key:

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

Where:

- "private_key.pem" is the private key generated in Step (a).
- "public_key.pem" is the file that contains the corresponding public key.

3. For each password that needs to be encrypted, do the following:

a. Type the password in plaintext in a text file using an editor. Don't hit enter at the end of the line. It should have just the password in a single line. In this example, the file is named as "pp1".

b. Execute:

```
openssl pkeyutl -encrypt -inkey public_key.pem -pubin -in pp1 -out encrypted_pp1
```

Where:

- "public_key.pem" is the public key generated in Step 2b.
- "pp1" is the file containing the single password in plaintext.
- "encrypted_pp1" contains the password in encrypted form.

Delete "pp1" created in *Step 3a* to avoid accidental exposure.

c. "encrypted_pp1" contains the key in raw binary form. Convert it to base64 as follows:

```
base64 encrypted_pp1
```

d. The above command (Step 3c) will output the base64 encoded encrypted password to the terminal. Copy and paste this into the "CUPSinfo.txt" file that contains the credentials supplied to the IOB tool. While copying, make sure to remove any line breaks or spaces. The entire password should be a single line.

e. "encrypted_pp1" can be deleted at this point.

Note *Step 3* must be performed for each password, one at a time, using the same public key/private key pair for all the passwords.

After "CUPSinfo.txt" file is updated with all the encrypted base64 passwords, the IOB tool is ready to be run. When running the script, specify an additional parameter: **-k** *absolute path to private_key.pem created in Step 2a>*.

Onboarding Application – Usage and Input Parameters

The application is compiled to create a standalone .exe. The application can be run on a RedHat Enterprise Linux machine.

The Onboarding Application can be run with below syntax:

```
./intelligent_onboarding -o <OP_Type_Parameter_File> -i <CUPS_Info_File> -k
<Path_to_Pvt_Key_file> [ -l <Path_to_store_logfiles> ] [ -p ] [ --context_selection_from_cp
 ] [ -v ]
```

Options:

- **-o**: [Mandatory] Provide the input parameter file specific to the operation being invoked.

  After successful onboarding, the IOB tool deletes the file.

- **-i**: [Mandatory] This option is used for "CUPSinfo.txt" file which has the details of CUPS system.

- **-k**: [Mandatory] Absolute path to the private key file. The tool uses this to decrypt the previously encrypted passwords. This private key file must correspond to the public key that is used to encrypt the passwords.

- **-p**: [Optional] When included, few pre-audit and post-audit checks are bypassed to reduce the time taken for Add/Modify/Delete operation.

- **-l**: [Optional] Provide absolute path to store the logs.

  When this keyword is not specified, the log files are created in the directory from which the IOB tool is invoked.

- **--context_selection_from_cp**: [Optional] When specified, the tool bases its context selection solely on the list of contexts available on the CP. The tool assumes that the selected context is also available on the UPs and does not validate this. This is an optimization. The default behavior is to examine contexts configured on CP and UP and select from contexts common to both.

- **-v**: [Optional] Displays the version of the IOB executable.

  If IOB tool is executed without the **-v** option, the version is displayed that is similar to:

  ```
  ##################################################################
  #                                                                #
  #             WELCOME TO ENTERPRISE ONBOARDING                   #
  #                   Version 21.20.9.private                      #
  #                                                                #
  ##################################################################
  ```

  NOTE: The version is displayed in the log file and terminal output as well.

CUPSinfo.txt

Onboarding application must know the system-level details to carry out the onboarding operations. The "CUPSinfo.txt" file has the IP addresses for CP and UP nodes and configurable threshold values. "Skip_UPGroup" and "Skip_Context" refers to the UP Groups and contexts that must not be considered for onboarding algorithm. For example, VIP groups and contexts that cannot be used for other enterprises. The file specifies a path where the configuration must be saved. The passwords in this file must be specified in RSA encrypted, base 64 format.

In 21.20.9 and earlier releases, the entry order of CP and UP inputs were:

```
//Control_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Primary_config_path,Secondary_config_path
```

```
//User_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Sx-IP-Address,Primary_config_path,Secondary_config_path
```

In 21.20.10 and later releases, the entry order of CP and UP inputs are:

```
//Control_Plane:
Host,Node,Primary_IP,Secondary-IP,Primary_config_path,Secondary_config_path,Login,Password
//User_Plane:
Host,Node,Primary_IP,Secondary-IP,Sx-IP-Address,Primary_config_path,Secondary_config_path,Login,Password
```

Sample CUPSinfo.txt File

For 21.20.9 and earlier releases:

```
//Threshold for Warning, input as percentage values

CPContext_threshold = {vrf_threshold:80, ipv4_threshold:80, ipv6_threshold:80}
CPSystem_threshold = {vrf_threshold:80, total_pool_threshold:80, apn_threshold:80}
UPContext_threshold = {vrf_threshold:80, ipv4_threshold:80, ipv6_threshold:80}
UPSystem_threshold = {vrf_threshold:80, apn_threshold:80, total_pool_threshold:80}
UPBudgeted_Sessions_threshold = {budgeted_threshold:80}

SKIP_UPGroup =
SKIP_Context =

//Control_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Primary_config_path,Secondary_config_path
cups_di_cp1,Control_Plane,209.165.200.225,209.165.200.225,<login_id>,<password>,
/flash/209.165.200.225-cups-vpp-saegw-global-control-plane.cfg,
/flash/209.165.200.225-cups-vpp-saegw-global-control-plane.cfg

//User_Plane:
Host,Node,Primary_IP,Secondary-IP,Login,Password,Sx-IP-Address,Primary_config_path,Secondary_config_path
cups_di_up0,User_Plane,209.165.200.230,209.165.200.230,<login_id>,<password>,
209.165.200.238,/flash/209.165.200.230-cups-vpp-saegw-global-user-plane-.cfg,
/flash/209.165.200.230-cups-vpp-saegw-global-user-plane.cfg
cups_di_up1,User_Plane,209.165.200.235,209.165.200.235,<login_id>,<password>,
209.165.200.242,/flash/209.165.200.235-cups-vpp-saegw-global-user-plane.cfg,
/flash/209.165.200.235-cups-vpp-saegw-global-user-plane.cfg
```

In 21.20.10 and later releases:

```
//Threshold for Warning, input as percentage values

CPContext_threshold = {vrf_threshold:98, ipv4_threshold:98,ipv6_threshold:98}
CPSystem_threshold = {vrf_threshold:98, total_pool_threshold:98, apn_threshold:98}
UPContext_threshold = {vrf_threshold:98, ipv4_threshold:98, ipv6_threshold:98}
UPSystem_threshold = {vrf_threshold:98, apn_threshold:98, total_pool_threshold:98}
UPBudgeted_Sessions_threshold = {budgeted_threshold:80}

SKIP_UPGroup =
SKIP_Context =

//Control_Plane: Host,Node,Primary_IP,Secondary-IP,Primary_config_path,
Secondary_config_path,Login,Password
cups_di_cp1,Control_Plane,209.165.200.225,209.165.200.225,/flash/209.165.200.225-CP01.cfg,
 /flash/209.165.200.225-CP02.cfg,<login_id>,<password>
//User_Plane: Host,Node,Primary_IP,Secondary-IP,Sx-IP-Address,Primary_config_path,
Secondary_config_path,Login,Password
cups_si_up1,User_Plane,209.165.200.235,209.165.200.235,209.165.200.242,/flash/209.165.200.235-UP01.cfg,
 /flash/209.165.200.235-UP02.cfg,<login_id>,<password>
```

ADD_ENTERPRISE_INPUT_PARAMETERS.txt

This file provides the configuration information when an APN is added. It provides the IP pool information and VRF information. The context provided is dummy and the actual context is determined as part of the algorithm. The IP pools doesn't support chunks.

Sample ADD_ENTERPRISE_INPUT_PARAMETERS.txt

Following is the example configuration for onboarding a single APN.

```
OpType = "ADD_ENTERPRISE"

CP_APN_Config  = '''Config
context APN
        apn starent.com
ip address pool name starent_ipv4_pool_group_01
ipv6 address prefix-pool starent_ipv6_pool_group_01
        exit
    exit
exit'''

// script will replace the dummy-SGI context with the chosen context
CP_SGi_Context = '''Config
    context dummy-SGi
ip vrf MPN00001
ip pool starent_ip_pool_v4_001 209.165.200.225 255.255.255.250 private 0 no-chunk-pool
group-name starent_ipv4_pool_group_01 vrf MPN00001
ip pool starent_ip_pool_v4_002 209.165.200.228 255.255.255.250 private 0 no-chunk-pool
group-name starent_ipv4_pool_group_01 vrf MPN00001

ipv6 pool starent_ip_pool_v6_001 prefix 2001:1:1::/48 private 0 no-chunk-pool group-name
starent_ipv6_pool_group_01 vrf MPN00001

    exit
exit'''

// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config= '''config
context dummy-SGI
ip vrf MPN00001
ip maximum-routes 100
exit
router bgp 65101
ip vrf MPN00001
route-distinguisher 65101 11100001
route-target both 65101 11100001
exit
address-family ipv4 vrf MPN00001
redistribute connected
exit
address-family ipv6 vrf MPN00001
redistribute connected
exit
exit
exit
exit'''
```

Following is the example configuration for onboarding multiple virtual APNs in one ADD operation.

```
OpType = "ADD_ENTERPRISE"

CP_APN_Config  = '''Config
```

```
context APN
apn virtual1
                ip address pool name apn2_ipv4_pool_group_01
                ipv6 address prefix-pool apn2_ipv6_pool_group_01
exit
apn virtual2
                ip address pool name apn2_ipv4_pool_group_02
                ipv6 address prefix-pool apn2_ipv6_pool_group_02
exit
apn virtual3
                ip address pool name apn2_ipv4_pool_group_03
                ipv6 address prefix-pool apn2_ipv6_pool_group_03
exit
apn virtual4
                ip address pool name apn2_ipv4_pool_group_04
                ipv6 address prefix-pool apn2_ipv6_pool_group_04
exit
apn virtual5
                ip address pool name apn2_ipv4_pool_group_05
                ipv6 address prefix-pool apn2_ipv6_pool_group_05
exit
apn virtual6
                ip address pool name apn2_ipv4_pool_group_06
                ipv6 address prefix-pool apn2_ipv6_pool_group_06
exit
apn virtual7
                ip address pool name apn2_ipv4_pool_group_07
                ipv6 address prefix-pool apn2_ipv6_pool_group_07
exit
apn virtual8
                ip address pool name apn2_ipv4_pool_group_08
                ipv6 address prefix-pool apn2_ipv6_pool_group_08
exit
apn virtual9
                ip address pool name apn2_ipv4_pool_group_09
                ipv6 address prefix-pool apn2_ipv6_pool_group_09
exit
apn virtual10
                ip address pool name apn2_ipv4_pool_group_10
                ipv6 address prefix-pool apn2_ipv6_pool_group_10
exit
apn real1
    virtual-apn preference 1 apn virtual2 domain virtual2
    virtual-apn preference 2 apn virtual3 domain virtual3
    virtual-apn preference 3 apn virtual4 domain virtual4
exit
apn real2
    virtual-apn preference 3 apn virtual5 domain virtual5
    virtual-apn preference 6 apn virtual6 domain virtual6
    virtual-apn preference 9 apn virtual7 domain virtual7
exit
apn real3
    virtual-apn preference 2 apn virtual6 domain virtual6
    virtual-apn preference 5 apn virtual7 domain virtual7
    virtual-apn preference 8 apn virtual8 domain virtual8
exit
apn real4
    virtual-apn preference 2 apn virtual8 domain virtual8
    virtual-apn preference 3 apn virtual9 domain virtual9
    virtual-apn preference 5 apn virtual10 domain virtual10
exit
apn real5
    virtual-apn preference 7 apn virtual10 domain virtual10
    virtual-apn preference 8 apn virtual1 domain virtual1
```

```
            virtual-apn preference 9 apn virtual2 domain virtual2
        exit
        apn real6
            virtual-apn preference 11 apn virtual10 domain virtual10
            virtual-apn preference 12 apn virtual1 domain virtual1
            virtual-apn preference 13 apn virtual2 domain virtual2
        exit
        apn real7
            virtual-apn preference 12 apn virtual2 domain virtual2
            virtual-apn preference 13 apn virtual3 domain virtual3
        exit
        apn real8
            virtual-apn preference 12 apn virtual7 domain virtual7
        exit
        apn real9
            virtual-apn preference 12 apn virtual5 domain virtual5
            virtual-apn preference 13 apn virtual6 domain virtual6
            virtual-apn preference 14 apn virtual7 domain virtual7
            virtual-apn preference 15 apn virtual8 domain virtual8
            virtual-apn preference 16 apn virtual9 domain virtual9
            virtual-apn preference 17 apn virtual10 domain virtual10
            virtual-apn preference 18 apn virtual2 domain virtual2
            virtual-apn preference 19 apn virtual3 domain virtual3
        exit
        apn real10
            virtual-apn preference 1 apn virtual1 domain virtual1
        exit
    exit
exit'''

// script will replace the dummy-SGI context with the chosen context
CP_SGi_Context = '''Config
    context dummy-SGi
        ip vrf MPN00002
        ip pool apn2_ip_pool_v4_001 209.165.201.1 255.255.255.224 private 0 group-name
apn2_ipv4_pool_group_01 vrf MPN00002 no-chunk-pool
        ip pool apn2_ip_pool_v4_002 209.165.201.3 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_02 vrf MPN00002
        ip pool apn2_ip_pool_v4_003 209.165.201.5 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_03 vrf MPN00002
        ip pool apn2_ip_pool_v4_004 209.165.201.7 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_04 vrf MPN00002
        ip pool apn2_ip_pool_v4_005 209.165.201.9 255.255.255.224 private 0 no-chunk-pool
group-name apn2_ipv4_pool_group_05 vrf MPN00002
        ip pool apn2_ip_pool_v4_006 209.165.201.11 255.255.255.224 private 0 no-chunk-pool
 group-name apn2_ipv4_pool_group_06 vrf MPN00002
        ip pool apn2_ip_pool_v4_007 209.165.201.13 255.255.255.224 private 0 no-chunk-pool
 group-name apn2_ipv4_pool_group_07 vrf MPN00002
        ip pool apn2_ip_pool_v4_008 209.165.201.15 255.255.255.224 private 0 no-chunk-pool
 group-name apn2_ipv4_pool_group_08 vrf MPN00002
        ip pool apn2_ip_pool_v4_009 209.165.201.17 255.255.255.224 private 0 no-chunk-pool
 group-name apn2_ipv4_pool_group_09 vrf MPN00002
        ip pool apn2_ip_pool_v4_010 209.165.201.19 255.255.255.224 private 0 no-chunk-pool
 group-name apn2_ipv4_pool_group_10 vrf MPN00002

        ipv6 pool apn2_ip_pool_v6_001 prefix 2001:268:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_01 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_002 prefix 2001:278:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_02 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_003 prefix 2001:288:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_03 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_004 prefix 2001:298:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_04 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_005 prefix 2001:2A8:1::/48 private 0 no-chunk-
```

```
pool group-name apn2_ipv6_pool_group_05 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_006 prefix 2001:2B8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_06 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_007 prefix 2001:2C8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_07 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_008 prefix 2001:2D8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_08 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_009 prefix 2001:2E8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_09 vrf MPN00002
        ipv6 pool apn2_ip_pool_v6_010 prefix 2001:2F8:1::/48 private 0 no-chunk-
pool group-name apn2_ipv6_pool_group_10 vrf MPN00002
    exit
exit'''

// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config   = '''config
                context dummy-SGI
                        ip vrf MPN00002
                                ip maximum-routes 100
                        exit
                        router bgp 65101
                                ip vrf MPN00002
                                        route-distinguisher 65101 11100002
                                        route-target both 65101 11100002
                                exit
                                address-family ipv4 vrf MPN00002
                                        redistribute connected
                                exit
                                address-family ipv6 vrf MPN00002
                                        redistribute connected
                                exit
                        exit
                exit
        exit'''
```

MODIFY_ENTERPRISE_INPUT_PARAMETERS.txt

This file provides the IP pools that must be added to or deleted from an existing enterprise. The context name is determined based on the pool name.

Sample MODIFY_ENTERPRISE_INPUT_PARAMETERS.txt

```
OpType = "MODIFY_ENTERPRISE"
CP_APN_Config  = '''Config
     context APN
            apn cisco.com
            exit
     exit
  exit'''
CP_SGi_Context = '''Config
    context dummy-SGi
     no ip pool cisco_ip_pool_v4_002 209.165.202.129 255.255.255.224 private 0 no-chunk-pool
 group-name starent_ipv4_pool_group_01 vrf MPN00001
     ip pool starent_ip_pool_v4_003 209.165.202.132 255.255.255.224 private 0 no-chunk-pool
 group-name starent_ipv4_pool_group_01 vrf MPN00001
    exit
exit'''
```

DELETE_ENTERPRISE_INPUT_PARAMETERS.txt

This input file must contain APN, SGi context, and VRF details when the request is to remove the enterprise.

Sample DELETE_ENTERPRISE_INPUT_PARAMETERS.txt

Following is the example configuration for deleting a single APN.

```
OpType= "DELETE_ENTERPRISE"

CP_APN_Config = '''config
     context APN
         no apn cisco.com
     exit
exit'''

// script will replace the dummy-SGI context with the chosen context
CP_SGi_Context = '''config
    context dummy-SGi
no ip vrf MPN00001

    exit
exit'''

// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config = '''config
      router bgp 65101
          no ip vrf MPN00001
      exit
exit'''
```

Following is the example configuration for deleting multiple virtual APNs in one DELETE operation.

```
CP_APN_Config = '''config
     context APN
         no apn virtual1
         no apn virtual2
         no apn virtual3
         no apn virtual4
         no apn virtual5
         no apn virtual6
         no apn virtual7
         no apn virtual8
         no apn virtual9
         no apn virtual10

         apn real1
            no virtual-apn preference 1
            no virtual-apn preference 2
            no virtual-apn preference 3
         exit
         apn real2
            no virtual-apn preference 3
            no virtual-apn preference 6
            no virtual-apn preference 9
         exit
         apn real3
            no virtual-apn preference 2
            no virtual-apn preference 5
            no virtual-apn preference 8
         exit
         apn real4
```

```
            no virtual-apn preference 2
            no virtual-apn preference 3
            no virtual-apn preference 5
         exit
         apn real5
            no virtual-apn preference 9
            no virtual-apn preference 8
            no virtual-apn preference 7
         exit
         apn real6
            no virtual-apn preference 13
            no virtual-apn preference 11
            no virtual-apn preference 12
         exit
         apn real7
            no virtual-apn preference 12
            no virtual-apn preference 13
         exit
         apn real8
            no virtual-apn preference 12
         exit
         apn real9
            no virtual-apn preference 19
            no virtual-apn preference 17
            no virtual-apn preference 13
            no virtual-apn preference 12
            no virtual-apn preference 15
            no virtual-apn preference 14
            no virtual-apn preference 16
            no virtual-apn preference 18
         exit
         apn real10
            no virtual-apn preference 1
         exit
     exit
exit'''


// script will replace the dummy-SGI context with the chosen context
CP_SGi_Context = '''config
    context dummy-SGi
        no ip vrf MPN00002

    exit
exit'''


// UP VRF config
// script will replace the dummy-SGI context with the chosen context
UP_VRF_Config = '''config
      router bgp 65101
          no ip vrf MPN00002
      exit
exit'''
```

System Limits

The following table depicts the maximum limits on ASR 5500 and CUPS.

Table 10: System Limits

Parameter	ASR 5500	Control Plane	User Plane
VRF Limit	300 per context 2048 per chassis	• 300 per context: Derived from the output of **show ip user-plane verbose** CLI command. • 1500 per chassis: Derived from the output of **show ip user-plane verbose** CLI command that is added across all contexts.	205 VRF (with default routes): Derived from the output of **show ip user-plane verbose** CLI command. Must calculate per UP.
IP Pool Limit	IPv4: 2000 per context IPv6: 256 IPv6 per context 5000 per chassis (combined IPv4 and IPv6)	IPv4: 2000 per context - Derived from the output of **show ip user-plane verbose** CLI command. IPv6: 256 IPv6 per context 3400 per chassis (combined IPv4 and IPv6) - Derived from the output of **show ip user-plane verbose** CLI command.	Total of 600 IP pools per context per UP group: • Total of 600 IP pools can consist a maximum of 256 IPv6 IP pools. • Total of 600 IP pools can consist a maximum of 600 IPv4 IP pools. Derived from the output of **show ip user-plane verbose** CLI command. Must calculate the value from the output (Max 600 IPv4 pools, Max 256 IPv6 pools).
APN Limit	2048	Total of 1500 for the system: Derived from the output of **show cups-resource session summary** CLI command.	205 per UP: Derived from the output of **show cups-resource session summary** CLI command. Must calculate per UP.

Note

- The IOB tool allows onboarding (OpType: ADD_ENTERPRISE) multiple APNs provided all those APNs share the "CP_SGi_Context" and "UP_VRF_Config" section of the input file. The APNs may potentially use different IP pool groups, but all those pool groups must be present in a single context in the "CP_SGi_Context" section of the input file. Also, the APNs must share a single VRF. In such a case all those APNs onboard to the same UP group and SGi context.
- The tool supports deletion (OpType: DELETE_ENTERPRISE) of multiple APNs provided that all APNs share the "UP_VRF_Config" and "CP_SGi_Context" sections of the input configuration. The tool deletes the VRF and pools at the end of the operation. The intended use case for multiple APN deletes is to delete APNs that were onboarded together. The APNs onboarded together must be deleted together - the tool does not support separate deletion of APNs that were onboarded together.
- The tool does not support modification (OpType: MODIFY_ENTERPRISE) of multiple APNs in one operation. Only one APN can be modified at a time.
- The CUPSinfo.txt file is considered as the primary UP information. If any UP Groups are added in the system, but are not present in the file, then they are excluded from onboarding.

Enterprise Onboarding in CUPS OAM Support

This section describes operations, administration, and maintenance information for this feature.

Show Commands

show cups-resource session summary

This CLI command is introduced in support of the Enterprise Onboarding in CUPS solution. The output of this CLI command displays system-level resources on CP.

NOTES:

- Group Name Column displayed in output is the name of UP Group.
- Sx-IP shows the IP address of UP configured under the UP Group.
- APN, Active-Sessions, and LCI details are for the UP Group.

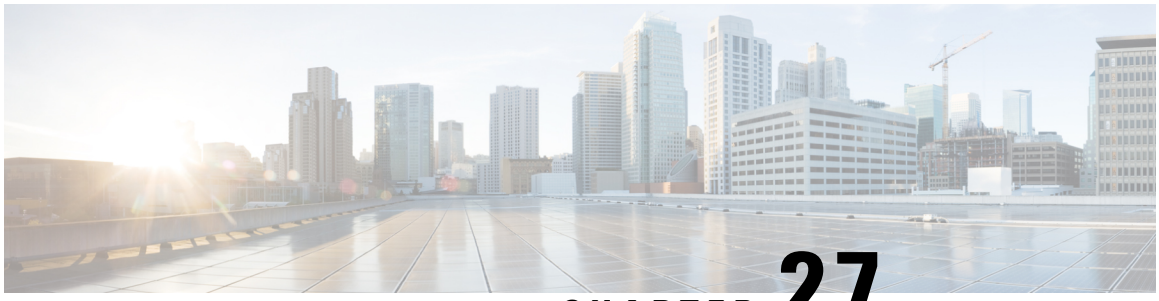
show ip user-plane verbose

The output of this CLI command is enhanced to display Total Pool Kernel Routes and Max Pool Kernel Routes fields. The dynamic IPv4 and IPv6 pool count is replaced with total IPv4 and IPv6 pool count. The output of this CLI command displays the context and UP Group it belongs to, and also adds information on number of IP pools and VRFs for that UP.

Error Codes

The following list of error codes is available in support of Enterprise Onboarding in CUPS feature.

Error Code	Description
1001	Indicates that the parsing of Input files has failed.
1002	Indicates that the parsing of Input_parameters file has failed.
1003	Indicates that the parsing of CUPSinfo file has failed.
1004	Indicates the inability to decrypt the passwords.
1005	Indicates that OpType is not present in input parameters.
1006	Indicates that the required configurations are not available in Input_parameters file for a given OpType.
1101	Indicates that the system pre-processing has failed.
1102	Indicates that the CPs pre-audit has failed for a given OpType.
1103	Indicates that the UPs pre-audit has failed for <UP_name>.
1107	Indicates that the tool is unable to update the CP_APN_Config section with the selected SGi context and UP Group. This indicates an error in the input configuration file.
1108	Indicates that the input file contains specified multiple APNs in a MODIFY_ENTERPRISE operation. This is not supported.
1301	Indicates that the CONTEXT and UPGROUP are not available for selection.
1401	Indicates the inability to find <context_name> and <group_name> from the CUPS system.
1501	Indicates the inability to get <context_name> from the output of **show apn** CLI command.
1601	Indicates that the configurations have failed for <control/user plane name> <connection state>.
1602	Indicates that the rollback configurations have failed for <control/user plane name> .
1701	Indicates that the CP post-audit has failed for <control plane name> <connection state>.
1702	Indicates that the UP post-audit has failed for <user plane name> <connection state>.
1703	Indicates that the Sx re-association has failed.



CHAPTER 27

Event-based CDRs for CUPS

This chapter includes the following topics:

- Revision History, on page 221
- Event-based CDRs for CUPS, on page 221
- Feature Description, on page 221
- How It Works, on page 222
- Standards Compliance, on page 224
- Monitoring and Troubleshooting, on page 224

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Event-based CDRs for CUPS

This chapter includes the following topics:

Feature Description

The CUPS architecture now supports Even-based Call Data Record (CDR) generation to account subscriber data usage. The EPC network, which consists of the User-Plane and Control-Plane as separate nodes, requires interaction between these entities to provide data usage accounting.

Generation of a CDR is an integral functionality of the Control-Plane. The Control-Plane interacts with the User-Plane to receive usage data such as: Uplink bytes, Downlink bytes and so on, to generate a CDR. These CDRs are generated based on Event Triggers. The event triggers can be either from the Access side of the

Control-Plane or PCRF generated. The usage data acquired from these events from the User-Plane, is updated in the CDR.

The following functionalities are supported in this feature:

- Exchange of Packet Flow Control Plane (PFCP) Session Modification Request and PFCP Session Modification Response messages.
- Reporting usage data from the User-Plane to the Control-Plane based on a configured Tariff-Time.

Note The scope of this feature is restricted only to P-GW and SAE-GW.

How It Works

The usage data report of a subscriber is retrieved from the User-Plane using the following mechanisms:

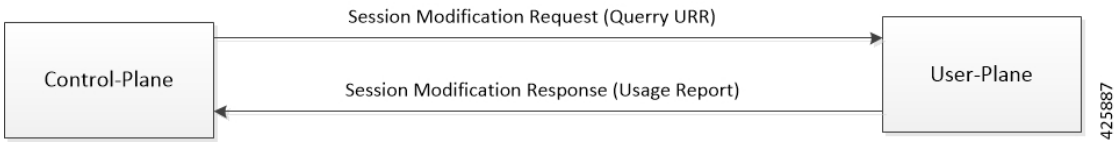
- **Pull mechanism**: The Control-Plane queries the User-Plane for the usage data report. The PFCP Session Modification Request/ PFCP Session Modification Response messages are used in this mechanism.
- **Push mechanism**: Here, the User-Plane sends the usage data report to the Control-Plane. The Tariff-Time configuration, which works with the existing Time/Volume-based push mechanism, is implemented. The PFCP Session Report Request/Session Report Response messages are used in this mechanism.

Fetching the Usage Report

In the CUPS architecture, because the User-Plane is a separate node, the Control-Plane node communicates with the User-Plane node using the PFCP protocol over the Sx interface to retrieve the usage data report of a subscriber.

The Control-Plane node sends the PFCP Session Modification request, containing the URRs for which the usage data report is reported. The User-Plane node responds with the PFCP Session Modification Response providing the usage data report for the requested URRs.

The following figure depicts the interaction between Control-Plane and User-Plane:



The following IEs are supported as part of the Sx Session Modification exchange messages:

- **Query URR**: This IE is present when the Control-Plane function requests immediate usage report(s) to the User-Plane function. Several IEs within the same IE type may be present to represent a list of URRs for which an immediate report is requested.
- **Usage Report**: This IE is present if the Query URR IE was present in the PFCP Session Modification Request and the traffic usage measurements for that URR are available at the User-Plane function. Several IEs within the same IE type may be present to represent a list of Usage Reports.

Tariff Time

Tariff-Time configuration is already supported by the Non-CUPS architecture. For CUPS, the Control-Plane uses the existing configuration. During a call set-up, PFCP Session Establishment Request carries the tariff time in the Monitoring Time IE, which is applicable to SDF URRs only. Bearer Level URR does not have this IE.

The Monitoring Time IE contains the configured time at which the usage data report of a subscriber is sent to the Control-Plane. Once the configured monitoring time expires the usage data report is sent, and sequentially, the time is automatically moved ahead by 24 hours indicating the time at which the next usage data report will be sent.

Note Before the next expiry of monitoring timer, usage data is reported continuously through the Time/Volume Threshold, if configured, or through an explicit request by the Control-Plane using the PFCP Session Modification Request (Query URR).

On the User-Plane, when the monitoring time expires for a URR, the Usage Report IE is sent to the Control-Plane. Sometimes, the monitoring time could expire for multiple subscribers at the same time. To avoid flooding of usage reports towards the Control-Plane, the User-Plane instead of reporting, piggybacks the usage data in the next outgoing message (PFCP Session Report Request or PFCP Session Modification Response) carrying the usage report.

The following IEs are supported as part of the Create URR IE within PFCP Session Modification Request:

- **Monitoring Time**: This IE contains the time at which the User-Plane function re-applies the volume or time threshold.
- **Subsequent Volume Threshold**: This IE may be present if the Monitoring Time IE is present and volume-based measurement is used. When present, it indicates the traffic volume value after which the User-Plane function reports the network resources usage to the Control-Plane function for the respective URR, for the period after the Monitoring Time.
- **Subsequent Time Threshold**: This IE may be present if the Monitoring Time IE is present and time-based measurement is used. When present, it indicates the traffic time value after which the User-Plane function reports the network resources usage to the Control-Plane function for the respective URR, for the period after the Monitoring Time.

Note In the non-CUPS architecture, P-GW supports four tariff-time instances in the Tariff-Time configuration. However, in CUPS only one tariff-time instance is supported.

Event Trigger

In this feature, an event trigger results in generation of either a partial CDR or a permanent CDR. In case of a partial event, only the CDR bucket is updated, but the actual CDR is not generated. But, in a permanent event trigger, a complete CDR is generated.

The following event triggers are supported in this feature:

- ULI Change (Partial event)

- Time Zone Change (Permanent event)
- Default Bearer QoS Change
- APN-AMBR Change

Note The GTPP trigger **egcdr max-losdv** is not supported in this release.

Standards Compliance

The Event-based CDRs for CUPS is based on the following standard(s):

- 3GPP TS 29.244: LTE; Interface between the Control Plane and the User Plane of EPC Nodes (3GPP TS 29.244 version 14.0.0 Release 14)

Monitoring and Troubleshooting

This section provides information on the show commands available to support Event-based CDRs for CUPS.

Show Commands and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature:

show active-charging subscribers full callid *call_id* urr-info

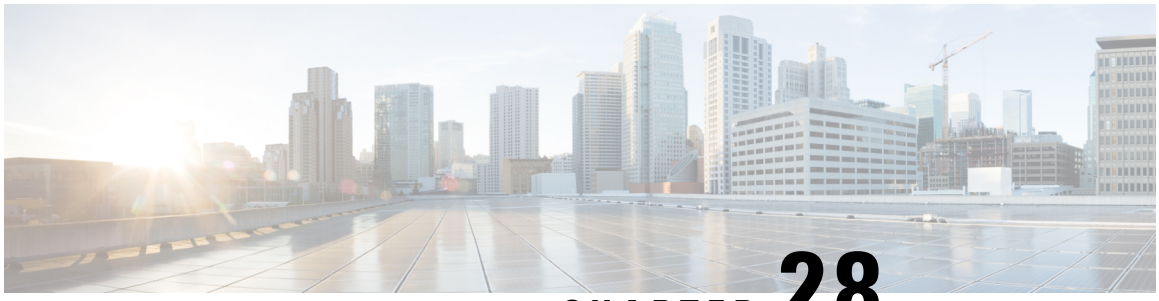
On executing the above command the following new fields are displayed:

- Next Monitoring Time
  - Subsequent Time Threshold
  - Subsequent Volume Threshold

show subscribers user-plane-only callid *call_id* urr full all

On executing the above command the following new fields are displayed:

- Next Monitoring Time
  - Subsequent Time Threshold
  - Subsequent Volume Threshold



CHAPTER 28

Event Data Records in CUPS

- Revision History, on page 225
- Feature Description, on page 225
- How It Works, on page 226
- Configuring Event Data Records in CUPS, on page 229
- Monitoring and Troubleshooting, on page 230

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
Support is added to generate interim EDRs.	21.23.6
First introduced	Pre 21.24

Feature Description

Generation of Event Data Records (EDR) is supported in the CUPS architecture.

These EDRs are generated on termination of a flow. Detailed information for every flow is generated after the flow termination.

Following are the EDR fields that gets populated in the event of EDR generation due to the flow end, transaction complete, and so on or whenever the necessary conditions are met.

- P2P Duration
- Rating group
- RADIUS NAS Identifier
- 3GPP Charging-id

- SN-Parent Protocol-id

When the data traffic with TCP starts for a subscriber attached to LTE network. Need to calculate and record time difference between control packets of TCP flow in EDR. Need to record the difference between following packets:

- SYN and SYN-ACK packet
- SYN-ACK and ACK packet

TCP Fast Open

TCP Fast Open (TFO) is an extension to speed up the opening of successive TCP connections between two endpoints. It works by using a TFO cookie (a TCP option), which is a cryptographic cookie stored on the client and set upon the initial connection with the server. When the client later reconnects, it sends the initial SYN packet along with the TFO cookie data to authenticate itself. If successful, the server may start sending data to the client even before the reception of the final ACK packet of the three-way handshake. Due to this RTT between SYN-ACK and ACK is calculated based on difference between SYN-ACK packet and first uplink ACK packet.

How It Works

EDRs are generated from UP on flow termination. During call setup and call modification, all call-specific attributes required for EDR generation is sent from CP to UP as part of the Subscriber Params IE within the Sx Establishment/Modification request messages.

On flow termination, the charging counters are fetched from VPP. All configured call-level attributes in the EDR format configuration along with the charging / volume counter attributes is sent to the CDRMOD proclet. This proclet writes these records to a file/disk, which is transferred to a configured external server.

Note User-Location-Information is written in hexadecimal format.

Transaction Complete EDR

Transaction complete EDRs are generated for HTTP EDRs when a HTTP transaction is complete. On completion, the charging counter are fetched from VPP. All configured call-level attributes in the EDR format configuration along with the charging / volume counter attributes is sent to the CDRMOD proclet. This proclet writes these records to a file/disk, which is transferred to a configured external server.

The following list of EDR attributes are supported:

- attribute sn-start-time
- attribute sn-end-time
- attribute sn-start-time format MM/DD/YYYY-HH:MM:SS:sss
- attribute sn-end-time format MM/DD/YYYY-HH:MM:SS:sss
- attribute radius-calling-station-id

- attribute radius-called-station-id
- rule-variable bearer 3gpp imsi
- rule-variable bearer 3gpp imei
- rule-variable bearer 3gpp rat-type
- rule-variable bearer 3gpp user-location-information
- rule-variable ip subscriber-ip-address
- rule-variable ip dst-address
- attribute sn-ruledef-name
- attribute sn-subscriber-port
- attribute sn-server-port
- attribute sn-app-protocol
- attribute sn-volume-amt ip bytes uplink
- attribute sn-volume-amt ip bytes downlink
- attribute sn-flow-start-time format seconds
- attribute sn-flow-end-time format seconds
- attribute sn-volume-amt ip pkts uplink
- attribute sn-volume-amt ip pkts downlink
- attribute sn-direction
- rule-variable traffic-type
- rule-variable p2p protocol
- rule-variable p2p app-identifier tls-cname
- rule-variable p2p app-identifier tls-sni
- rule-variable p2p app-identifier quic-sni
- rule-variable bearer 3gpp sgsn-address
- attribute sn-rulebase
- attribute sn-charging-action
- rule-variable flow tethered-ip-ttl
- rule-variable flow ttl
- rule-variable flow ip-control-param
- rule-variable bearer qci
- rule-variable tcp flag
- rule-variable ip server-ip-address

- attribute sn-flow-id
- attribute sn-closure-reason
- attribute sn-duration
- rule-variable ip src-address
- rule-variable ip protocol
- attribute sn-charge-volume ip bytes uplink
- attribute sn-charge-volume ip bytes downlink
- tcp-state
- tcp-prev-state

The following HTTP EDR attributes are supported:

- rule-variable http url length 2000
- rule-variable http request method
- rule-variable http content type
- rule-variable http user-agent length 255
- rule-variable http reply code
- rule-variable http referer
- rule-variable http host
- rule-variable http cookie
- rule-variable http header-length
- attribute transaction-uplink-bytes
- attribute transaction-downlink-bytes

Support for Interim EDRs

ECS supports generation of Interim EDRs – EDRs that are generated for ongoing flows based on a configurable timer.

Usually, EDRs are generated for flows only when the flow terminates or when the flow reaches the configured flow idle-timeout value. These flows could have time duration that is as long as 48 hours, which makes it difficult to track subscriber activity until an EDR is generated.

Thus, with interim EDRs, ongoing flow activities are tracked by configuring an interim timeout value for a flow. On expiration of the interim timer, an EDR is generated.

For configuring an interim EDR, a new CLI keyword, **interim**, is introduced. Based on the configuration, the interim timer is applied to newly created flows. On expiration of the timer, an interim EDR is generated along with the following reason: **sn-closure-reason (23)**. The information volume available until the expiration of the timer is populated in the EDR along with its respective timestamps.

Limitations

The Event Data Record feature in CUPS has the following limitations:

- EDR will be generated only for flow end condition – idle timeout, hagr, normal flow termination & during end of session.
- Charging-Action based EDR configuration is not supported.
- Reporting EDRs are not supported.

Configuring Event Data Records in CUPS

Configuration on CP to Push EDRs to UP

Use the following configuration to push EDRs from CP to UP using PFD mechanism.

Note The CLI commands used in this configuration are part of the existing non-CUPS architecture.

```
active-charging service service_name
   rulebase rulebase_name
      flow end-condition { timeout | normal-end-signaling | session-end |
 interim  } charging-edr charging_edr_format_name
      edr transaction-complete http charging-edr charging_edr_format_name
      exit
      edr-format format_name
         attribute attribute_name
         end
```

NOTES:

- **flow end-condition**: This command allows you to configure the end condition of the session flows related to a user session and triggers EDR generation.
- **timeout**: Creates an EDR with the specified EDR format whenever a flow ends due to a timeout condition.
- **normal-end-signaling**: Creates an EDR with the specified EDR format whenever flow end is signaled normally.
- **session-end**: Creates an EDR with the specified EDR format whenever a subscriber session ends. By this option session manager creates an EDR with the specified format name for every flow that has had any activity since last EDR was created for the flow on session end.
- **charging-edr** *charging_edr_format_name*: Specifies the charging EDR format.
- **interim**: This condition specifies the interim threshold condition of the flow where an EDR is generated based on the configured timer value. The *interim_timer_value* is configured in minutes with a configurable range from 15 to 1440 minutes.
- The **interim** keyword is only applicable for new flows created and not on existing flows.

- **http**: Specifies HTTP protocol related configuration.

Configuration to Enable EDR Module on UP

Use the following configuration to enable EDR module on UP

Note The CLI commands used in this configuration are part of the existing non-CUPS architecture.

```
configure
   context context_name
     edr-module active-charging-service
     end
```

Configuring Additional TCP Fields

Prior to using the following CLI commands to configure additional TCP fields in the EDR, ensure that all the other EDR configurations are present.

Note For CUPS setup, once configuration is present on CP side, push those changes on UP using **push config-to-up all** command from CP.

```
configure
   active-charging service service_name
     edr-format edr_format_name
        [ no ] rule-variable tcp syn_synack_rtt priority 3
        [ no ] rule-variable tcp syn_synack_ack_rtt priority 4
        end
```

Monitoring and Troubleshooting

show user-plane-service statistics rulebase name *rulebase_name*

The following fields are displayed in support of this feature:

- Rulebase Name
  - EDRs
  - Charge Volume
    - Uplink Pkts
    - Uplink Bytes
    - Downlink Pkts

- Downlink Bytes

- Charging EDRs
  - Total Charging EDRs generated
  - EDRs generated for handoff
  - EDRs generated for timeout
  - EDRs generated for normal-end-signaling
  - EDRs generated for session end
  - EDRs generated for rule match
  - EDRs generated for hagr
  - EDRs generated for flow-end content-filtering
  - EDRs generated for flow-end url-blacklisting
  - EDRs generated for content-filtering
  - EDRs generated for url-blacklisting
  - EDRs generated for any-error packets
  - EDRs generated for firewall deny rule match
  - EDRs generated for transaction completion
  - EDRs generated for voip call end
  - EDRs generated for dcca failure handling
  - EDRs generated for TCP optimization on
  - EDRs generated for tethering signature change
  - EDRs generated for interim interval
  - Total Flow-Overflow EDRs
  - Total zero-byte EDRs suppressed

- EDRs generated for interim
  - Interval

- Total Rulebases

show active-charging rulebase statistics real-time

The following fields are displayed in support of this feature:

- Rulebase Name
- Charging EDRs

- Total Charging EDRs generated
  - EDRs generated for handoff
  - EDRs generated for timeout
  - EDRs generated for normal-end-signaling
  - EDRs generated for session end
  - EDRs generated for rule match
  - EDRs generated for hagr
  - EDRs generated for flow-end content-filtering
  - EDRs generated for flow-end url-blacklisting
  - EDRs generated for content-filtering
  - EDRs generated for url-blacklisting
  - EDRs generated for any-error packets
  - EDRs generated for firewall deny rule match
  - EDRs generated for transaction completion
  - EDRs generated for voip call end
  - EDRs generated for dcca failure handling
  - EDRs generated for TCP optimization on
  - EDRs generated for tethering signature change
  - EDRs generated for interim interval
  - EDRs generated for audio-end Sessions
  - EDRs generated for video-end Sessions
  - EDRs generated for voipout-end Sessions
  - Total Flow-Overflow EDRs
  - Total zero-byte EDRs suppressed

show active-charging edr-format all

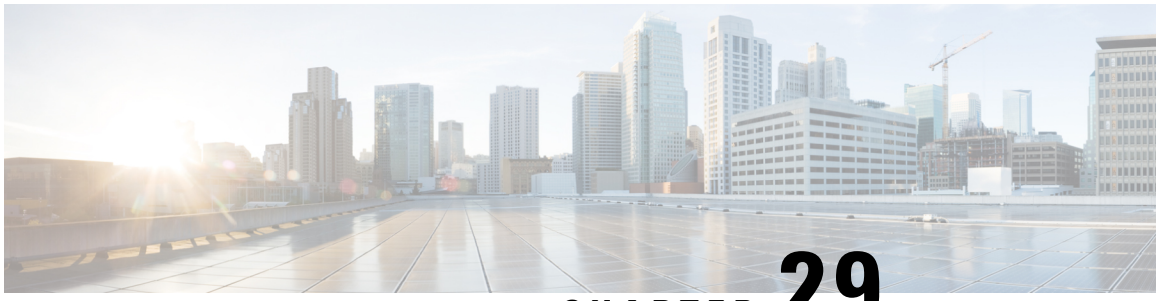
The following fields are displayed in support of Additional TCP Fields in EDR feature:

- Service Name
  - EDR Format Name
    - rule-variable tcp syn-synack-rtt priority 3
    - rule-variable tcp synack-ack-rtt priority 4

Bulks Statistics

The following bulk statistic(s) are added in the ECS schema to support Event Data Records in CUPS:

- edrs-generated: Indicated the total number of EDRs generated.



CHAPTER 29

Error Indication and GTPU Path Failure Detection

- Revision History, on page 235
- Feature Description, on page 235
- How It Works, on page 236
- Configuring Error Indication and GTPU Path Failure on Control Plane, on page 242

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The User Plane (UP) function notifies an Error Indication message for a GTPU peer to the sender when a GTP-PDU is received with a TEID that does not exist. This ensures that there are no stale sessions or bearers and maintains consistency in the network.

Error Indication and GTPU Path Failure between CP and UP nodes are supported over SxA, SxB and SxAB. For the neighbor nodes, it is supported over the S1u/S5u interfaces.

Behavior variations of local-purge or signal-peer for error indication and GTPU path failure are considered in this implementation.

- When Error Indication is received, the UP communicates the TEID and GTPU-peer information with the CP to ensure deletion or modification of the GTPU-peer.
- On receiving GTPU packet with non-existing TEID, the UP generates and sends Error Indication with TEID and GTPU peer entries.
- The deletion of a session or a bearer is decided based on the path failure detection at CP or UP.

- GTPU path failure is detected using GTPU echo messages between UP nodes, and between the UP and CP nodes.

As per 3GPP TS 29.244, the following is implemented in this feature:

- The PFCP Session Report Request is sent over the Sxa and Sxb interface by the UP function to report information related to an PFCP session to the CP function.
- The PFCP Session Report Response is sent over the Sxa and Sxb interface by the CP function to the UP function as a response to the PFCP Session Report Request.
- Error Indication Report IE must be present if the Report Type indicates an Error Indication Report.
- Remote F-TEID is sent in the Error Indication Report to identify the remote F-TEID of the GTP-U bearer for which an Error Indication has been received at the UP function.
- The PFCP Node Report Request is sent over the Sxa and Sxb interface by the UP function to report information to the CP function that is not specific to an PFCP session.
- The PFCP Node Report Response is sent over the Sxa, Sxb; Sxc and N4 interface by the CP function to the UP function as a response to the PFCP Node Report Request.
- UPPath Failure Report will be present if the Node Report Type indicates a User Plane Path Failure Report.
- Remote GTP-U Peer includes the IP address of the remote GTP-U peer towards which a UP path failure has been detected.

How It Works

Error Indication Support

Error Indication Handling at CP

CP on receiving a PFCP Session Report Request triggered by Error Indication received on UP from a neighboring UP, responds with PFCP Session Report Response and sends a PFCP Session Modification Request towards UP to delete PDR, a FAR for dedicated bearer identified for removal or a PFCP Session Deletion Request to delete the session.

- The session or bearer will be locally purged on PGW-C on reception of PFCP Session Deletion Response or PFCP Session Modification Response from UP respectively.
- For SAEGW-C, signaling over EGTP is based on **local purge** and **page-ue** configuration for S1u.
- For SGW-C, signaling over EGTP on CP is based on **local purge** and **page-ue** configuration for S1u and local-purge and signal peer on S5u.

Error Indication Handling on UP

UP on receiving Error Indication initiates a PFCP Session Report Request with Error Indication Report that includes remote FTEID containing TEID and GTPU Peer address.

- For PGW-U, Error Indication messages is sent or received over S5u.

- For SAEGW-U, Error Indication message is sent or received over S1u.
- For SGW-U, Error Indication message is sent and received over S1u and S5u.

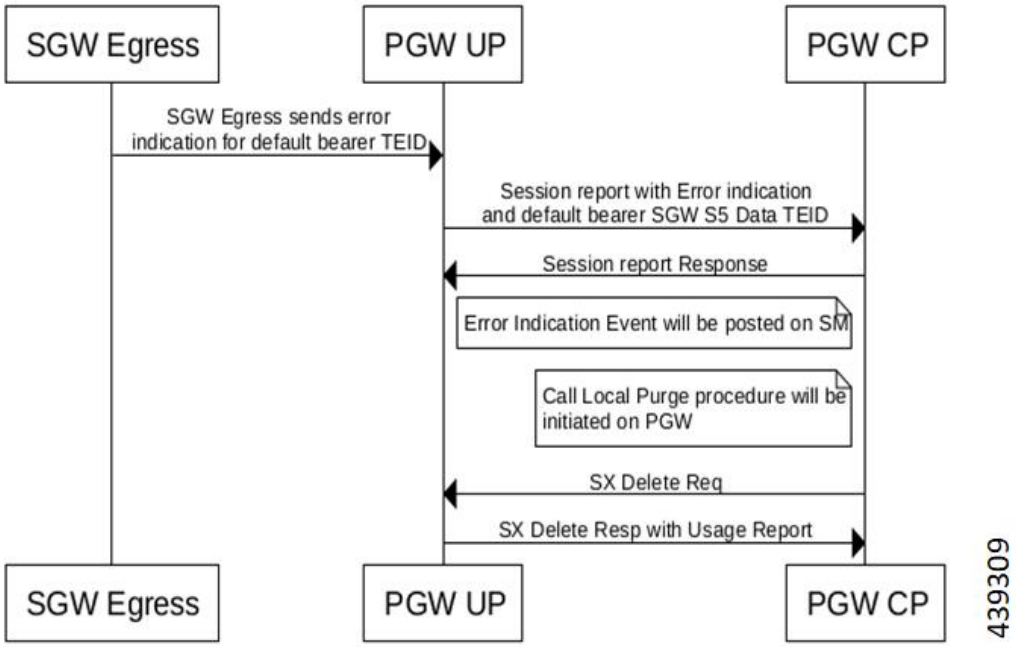
Error Indication Generation on UP

UP generates Error Indication with TEID and GTPU Peer Address towards a peer when a data packet is received with TEID for which a session or bearer does not exist.

Error Indication Call Flows

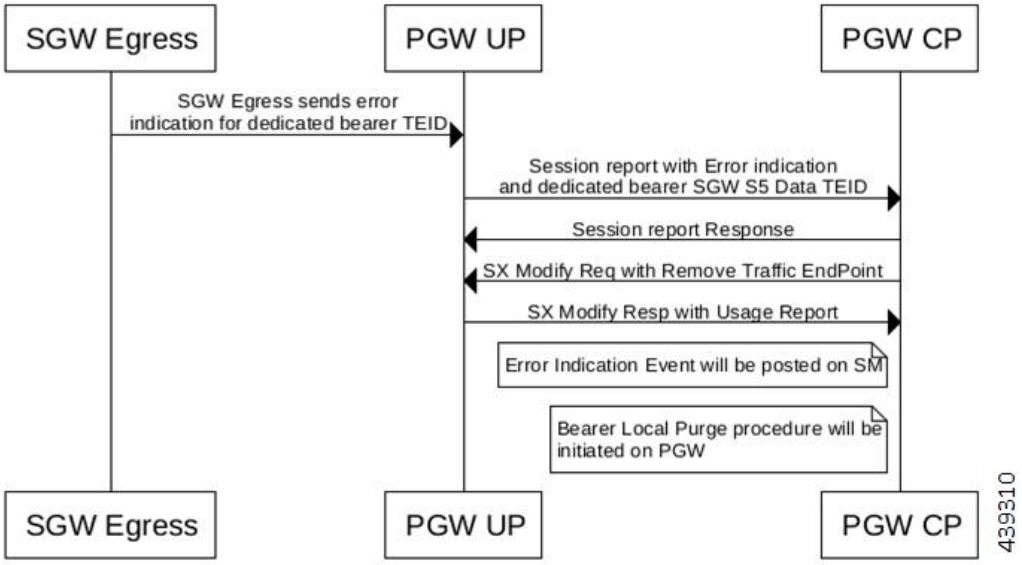
P-GW Default Bearer Error Indication Handling

The following call flow illustrates P-GW default bearer error indication handling with local purge.



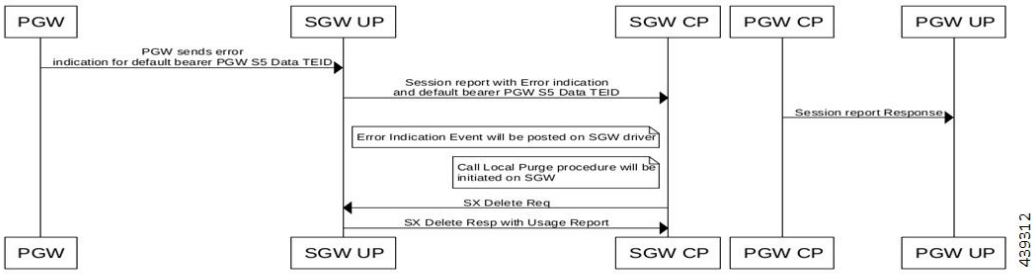
P-GW Dedicated Bearer Error Indication Handling

The following call flow illustrates P-GW dedicated bearer error indication handling with local purge.



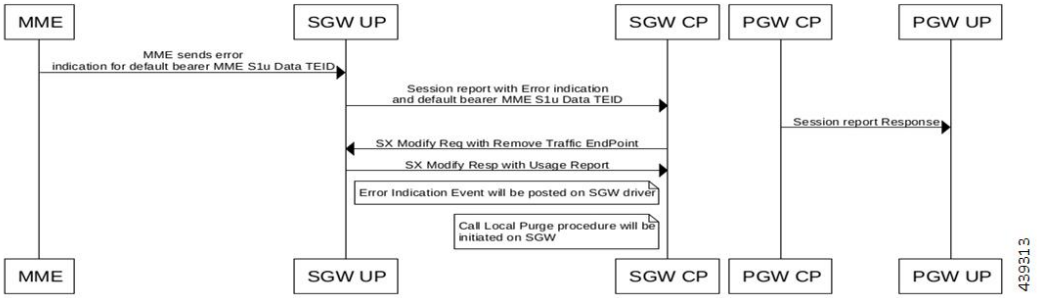
S-GW Default Bearer Indication Handling

The following call flow illustrates S-GW dedicated bearer error indication handling with S5u local purge.



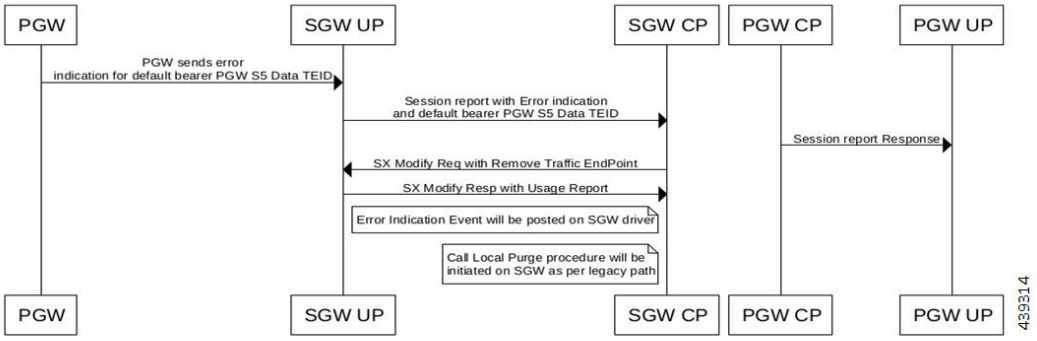
S-GW Dedicated Bearer Indication Handling

The following call flow illustrates S-GW dedicated bearer error indication handling with S1u local purge.



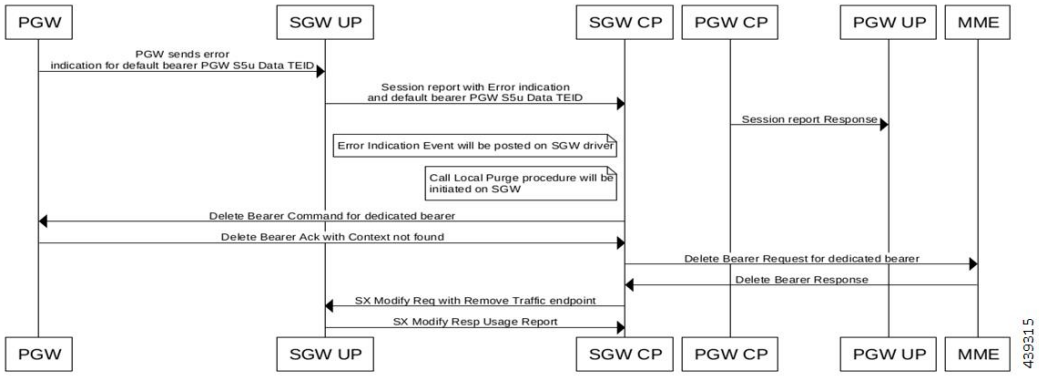
S-GW Dedicated Bearer Indication Handling

The following call flow illustrates S-GW dedicated bearer error indication handling with S5u local purge.



S-GW Dedicated Bearer Indication Handling

The following call flow illustrates S-GW dedicated bearer error indication handling with S5u signal peer.



GTPU Path Failure Support

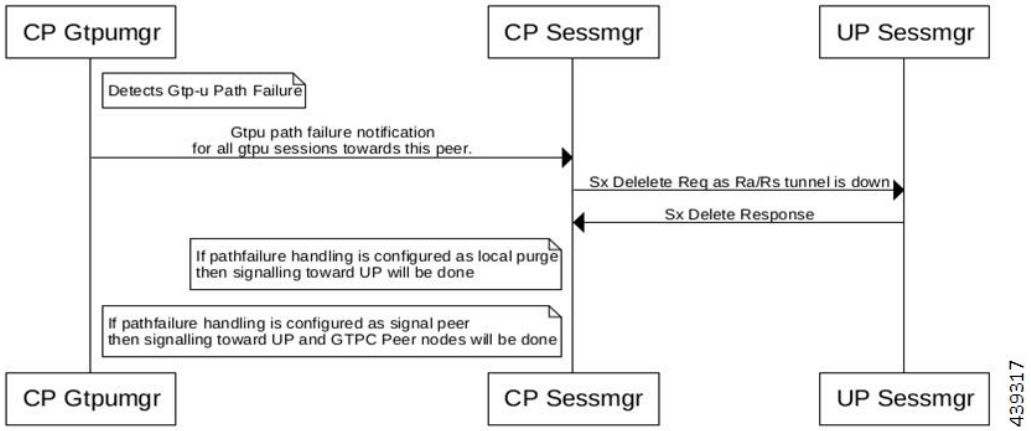
GTPU Path Failure Support at CP

GTPU Echo Requests is initiated and sent periodically as per the configured interval on CP. GTPU Echo Response is sent for the GTPU Echo Request received from UP over the GTPU tunnel.

If Response is not received for the GTPU Echo Request, CP retries Echo Requests based on configured retransmission timeout and maximum retries. When retries are exhausted, CP initiates PFCP Session Deletion Request to delete the PFCP session.

On receiving the PFCP Node Report Request from UP, CP will send PFCP Node Report Response and initiate PFCP Session Deletion Request towards UP. Billing records will be generated when usage reports are received in PFCP Session Deletion Response.

The following call flow illustrates GTPU Path Failure handling at CP.



GTPU Path Failure Support at UP

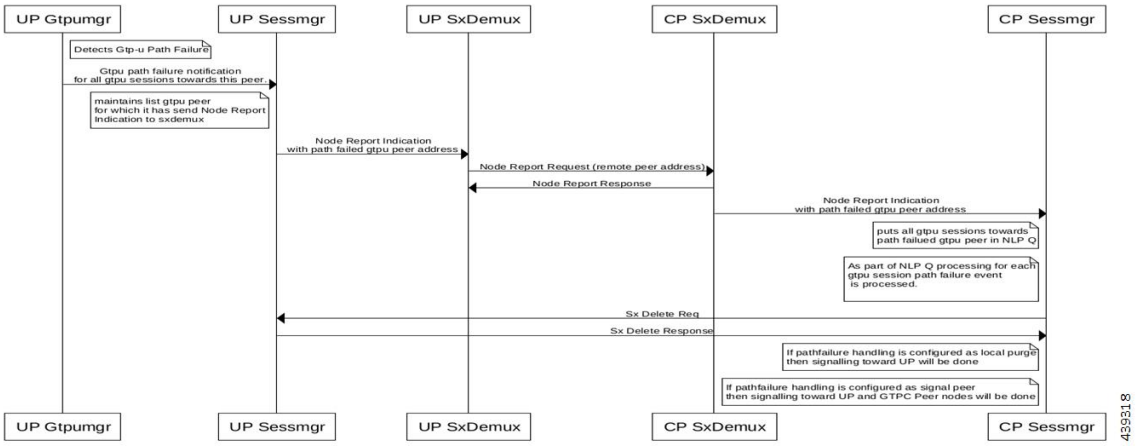
GTPU Echo Requests is initiated and sent periodically as per the configured interval on UP. GTPU Echo Response is sent for the GTPU Echo Request received from CP over GTPU tunnel.

If Response is not received for the GTPU Echo Request, UP retries Echo Requests based on configured retransmission timeout and maximum retries. When retries are exhausted, UP shall initiate PFCP Node

Report Request including (Node ID, Node Report Type, User Plane Path Failure Report including Remote GTP-U Peer).

If UP receives PFCP Node Report Response and PFCP Session Deletion Request to delete the session, it responds to the deletion request with usage reports.

The following call flow illustrates GTPU Path Failure support at UP



Limitations

In this release, the Error Indication and GTPU Path failure feature has the following limitations:

- UP on receiving following messages/packets with Extension Headers will respond with Supported Extension Headers Notification indicating neighboring UPs that extension headers are not supported.
  - Error Indication
  - GTPU Echo Requests
  - GTPU Echo Response
  - GTP-PDU

Configuring Error Indication and GTPU Path Failure on Control Plane

Configuring Error Indication on CP

Use following commands to control the behavior of CP towards EGTP peers based on GTPU error indication received on a GTPU interface (s1u/s5u).

```
configure
  context context_name
    sgw-service service_name
```

```
      gtpu-error-ind { s1u { local-purge | page-ue }| s5u { local-purge
| signal-peer } }
      end
```

NOTES:

- **gtpu-error-ind**: Configures the actions to be taken upon receiving a GTP-U error indication from P-GW.
- **s1u**: Specifies the action to take when a GTP-U error indication is received from P-GW over the S1u interface.
- **s5u**: Specifies the action to take when a GTP-U error indication is received from P-GW over the S5u interface.
- **local-purge**: The S-GW clears the affected bearer (or PDN if error-indication is received on default bearer) locally without informing peer.
- **page-ue**: The S-GW moves the complete UE state to S1-Idle and starts paging for this UE.
- **signal-peer**: Clears the affected bearers or PDNs and initiates control signals towards the peer MME and P-GW.

Note The **extension-header source-udp-port** CLI option is not supported for GTP-U service on User Plane.

Configuring GTPU Path Failure on CP

Use following commands to control the behavior of CP towards EGTP peers based on GTPU path failure detected on GTPU interface (s1u/s5u).

```
configure
  context context_name
    sgw-service service_name
      path-failure { s1u | s5u }{ local-purge | signal-peer }
      end
```

NOTES:

- **path-failure**: Configures the action to take upon the occurrence of a path failure between the S-GW and the MME or P-GW.
- **s1u**: Specifies the action to take when a GTP-U error indication is received from P-GW over the S1u interface.
- **s5u**: Specifies the action to take when a GTP-U error indication is received from P-GW over the S5u interface.
- **local-purge**: The S-GW clears the affected bearer (or PDN if error-indication is received on default bearer) locally without informing peer.
- **signal-peer**: Clears the affected bearers or PDNs and initiates control signals towards the peer MME and P-GW.

Limitations

The following CLI options are not supported in this release:

- **In GTP-U service on UP: extension-header source-udp-port**
- **In SG-W service on CP:**

  gtpu-error-ind s4u

  gtpu-error-ind s11u

  gtpu-error-ind s12

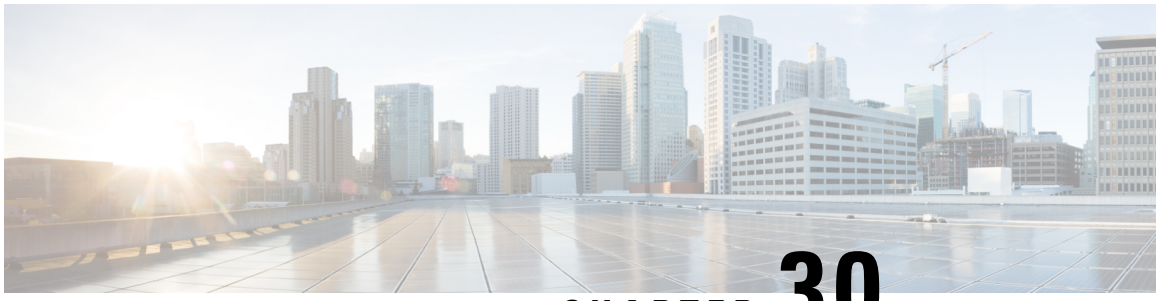
  path-failure s4u

  path-failure s11u

  path-failure s12

When Sx Session Modification Response for Error Indication or GTP-U Path Failure is pending from User Plane and Collapsed to Pure-P Handover request is received, Modify Bearer Request for Handover is processed once Sx Session Modification Response which was delayed is received. Following configuration is recommended for working of above case for handover to be successfully completed.

```
configure
   context egress context_name
      ims-auth-service service_name
         policy-control
            max-outstanding-ccr-u 2
            end
```



CHAPTER 30

Firewall Support in CUPS

- Revision History, on page 245
- Feature Description, on page 245
- Configuring the Default Firewall Feature, on page 246
- Monitoring and Troubleshooting, on page 248
- Show CLIs for CUPS, on page 249
- SNMP Traps, on page 249
- Reassembly Behavior Change, on page 250

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Subscriber Firewall feature on CUPS architecture allows you to configure Stateless and Stateful packet inspection and packet filtering to protect the subscribers from malicious attacks. The firewall configuration allows the system to inspect each packet of the subscriber data session. It also evaluates the security threat and applies the policies configured on uplink and downlink traffic.

Note The subscriber firewall implementation in CUPS is like the firewall implementation in non-CUPS architecture. For more details on the subscriber firewall in non-CUPS, see the *PSF Administration Guide*.

Overview

Firewall feature includes the support for the following:

- DoS attack
- DDoS attack
- Packet Filtering
- Stateless & stateful packet inspection
- Application level gateways
- SNMP thresholding and logging

Configuring the Default Firewall Feature

Following is the default configuration for the FW policy.

```
configure
          active-charging service service_name
            fw-and-nat policy policy_name
            end
```

Along with the preceding service configuration, Following is the default CLI behavior of various FW related CLI within the service.

```
Dos-Protection:
      Source-Route                        : Disabled
      Win-Nuke                            : Disabled
      Mime-Flood                          : Disabled
      FTP-Bounce                          : Disabled
      IP-Unaligned-Timestamp              : Disabled
      Seq-Number-Prediction               : Disabled
      TCP-Window-Containment              : Disabled
      Teardrop                            : Disabled
      UDP Flooding                        : Disabled
      ICMP Flooding                       : Disabled
      SYN Flooding                        : Disabled
      Port Scan                           : Disabled
      IPv6 Extension Headers Limit        : Disabled
      IPv6 Hop By Hop Options             : Disabled
      Hop By Hop Router Alert Option      : Disabled
      Hop By Hop Jumbo Payload Option     : Disabled
      Invalid Hop By Hop Options          : Disabled
      Unknown Hop By Hop Options          : Disabled
      IPv6 Destination Options            : Disabled
      Invalid Destination Options         : Disabled
      Unknown Destination Options         : Disabled
      IPv6 Nested Fragmentation           : Disabled

    Max-Packet-Size:
      ICMP                          : 65535
      Non-ICMP                      : 65535
    Flooding:
      ICMP limit                    : 1000
      UDP limit                     : 1000
      TCP-SYN limit                 : 1000
```

```
  Sampling Interval              : 1

TCP-SYN Flood Intercept:
  Mode                           : None
  Max-Attempts                   : 5
  Retrans-timeout                : 60
  Watch-timeout                  : 30
Mime-Flood Params:
  HTTP Header-Limit              : 16
  HTTP Max-Header-Field-Size     : 4096

No Firewall Ruledef Match Action:
  Uplink Action                  : permit
  Downlink Action                : deny

TCP RST Message Threshold     : Disabled
ICMP Dest-Unreachable Threshold    : Disabled
Action upon receiving TCP SYN packet with ECN/CWR Flag set    : Permit
Action upon receiving a malformed packet  : Deny
Action upon IP Reassembly Failure   : Deny
Action upon receiving an IP packet with invalid Options : Permit
Action upon receiving a TCP packet with invalid Options : Permit
Action upon receiving an ICMP packet with invalid Checksum: Deny
Action upon receiving a TCP packet with invalid Checksum: Deny
Action upon receiving an UDP packet with invalid Checksum: Deny
Action upon receiving an ICMP echo packet with id zero : Permit
TCP Stateful Checks : Enabled
First Packet Non-SYN Action: Drop
ICMP Stateful Checks: Enabled
TCP Partial Connection Timeout: 30
```

Enabling Firewall for IPv4 and IPv6

Following is the configuration to enable the firewall for IPv4 and IPv6:

```
configure
          active-charging service service_name
             fw-and-nat policy policy_name
                firewall policy ipv4-and-ipv6
                end
```

Configuration Support for Subscriber Firewall

The Control Plane pushes the required configuration for the subscriber firewall to the User Plane through PFD management. Firewall configurations are available under active charging configuration.

- Access-Rule-Defs
- Firewall-Nat Policy

Firewall feature configuration supports activation of firewall feature using rulebase, APN-based, and/or subscriber-based activation.

This section details the different aspect of configuration for the subscriber firewall in CUPS.

- Config delete command deletes the configuration immediately. It doesn't wait for bulk config timer as the said config is removed from the SCT and it's deleted from all Sessmgrs immediately.

- Addition/deletion/Modification of firewall configuration from CP to UP propagates using CLI command "push config-to-up all".

Monitoring and Troubleshooting

Following is the show command output for the default Firewall feature on Control Plane.

show config active-charging service name acs verbose

```
fw-and-nat policy SFW_NAT_TEST
      no firewall dos-protection source-router
      no firewall dos-protection winnuke
      no firewall dos-protection mime-flood
      no firewall dos-protection ftp-bounce
      no firewall dos-protection ip-unaligned-timestamp
      no firewall dos-protection tcp-window-containment
      no firewall dos-protection teardrop
      no firewall dos-protection flooding udp
      no firewall dos-protection flooding icmp
      no firewall dos-protection flooding tcp-syn
      no firewall dos-protection port-scan
      no firewall dos-protection ipv6-extension-hdrs
      no firewall dos-protection ipv6-hop-by-hop
      no firewall dos-protection ipv6-hop-by-hop router-alert
      no firewall dos-protection ipv6-hop-by-hop jumbo-payload
      no firewall dos-protection ipv6-hop-by-hop invalid-options
      no firewall dos-protection ipv6-hop-by-hop unknown-options
      no firewall dos-protection ipv6-dst-options
      no firewall dos-protection ipv6-dst-options invalid-options
      no firewall dos-protection ipv6-dst-options unknown-options
      no firewall dos-protection ipv6-frag-hdr nested-fragmentation
      no firewall dos-protection ip-sweep tcp-syn
      no firewall dos-protection ip-sweep udp
      no firewall dos-protection ip-sweep icmp
      firewall max-ip-packet-size 65535 protocol icmp
      firewall max-ip-packet-size 65535 protocol non-icmp
      firewall flooding protocol icmp packet limit 1000
      firewall flooding protocol udp packet limit 1000
      firewall flooding protocol tcp-syn packet limit 1000
      firewall flooding sampling-interval 1
      firewall tcp-syn-flood-intercept mode none
      firewall tcp-syn-flood-intercept watch-timeout 30
      firewall mime-flood http-headers-limit 16
      firewall mime-flood max-http-header-field-size 4096
      no firewall icmp-destination-unreachable-message-threshold
      access-rule no-ruledef-matches uplink action permit
      access-rule no-ruledef-matches downlink action deny
      firewall tcp-idle-timeout-action reset
      no firewall tcp-reset-message-threshold
      firewall tcp-syn-with-ecn-cwr permit
      firewall malformed-packets drop
      firewall ip-reassembly-failure drop
      no firewall validate-ip-options
      firewall tcp-options-error permit
      firewall icmp-echo-id-zero permit
      firewall icmp-checksum-error drop
      firewall tcp-checksum-error drop
      firewall udp-checksum-error drop
      firewall tcp-fsm first-packet-non-syn drop
      firewall icmp-fsm
```

```
  firewall policy ipv4-and-ipv6
  firewall tcp-partial-connection-timeout 30
  no nat policy
  no nat binding-record
  no nat pkts-drop edr-format
  no nat pkts-drop timeout
  default nat suppress-aaa-update
  nat private-ip-flow-timeout 180
  nat check-point-info basic limit-flows 100
  nat check-point-info sip-alg
  nat check-point-info h323-alg
  nat max-chunk-per-realm single-ip
#exit
```

Show CLIs for CUPS

Following are the show CLIs for the CUPS:

For User Plane:

- show subscribers user-plane-only full all
- show subscribers user-plane-only flows
- show user-plane-service inline-services firewall statistics verbose
- show user-plane-service statistics rulebase all
- show alarm outstanding all
- show alarm outstanding all verbose
- show alarm statistics
- show user-plane-service statistics rulebase name <rulebasename>

For Control Plane:

- show active-charging fw-and-nat policy all
- show active-charging fw-and-nat policy name "fw_nat_policy_name"
- show active-charging firewall track-list attacking-servers
- show active-charging ruledef name

SNMP Traps

Following are the SNMP traps in support of this feature for CUPS, Use the respective trap CLIs on the User Plane to enable the trap.

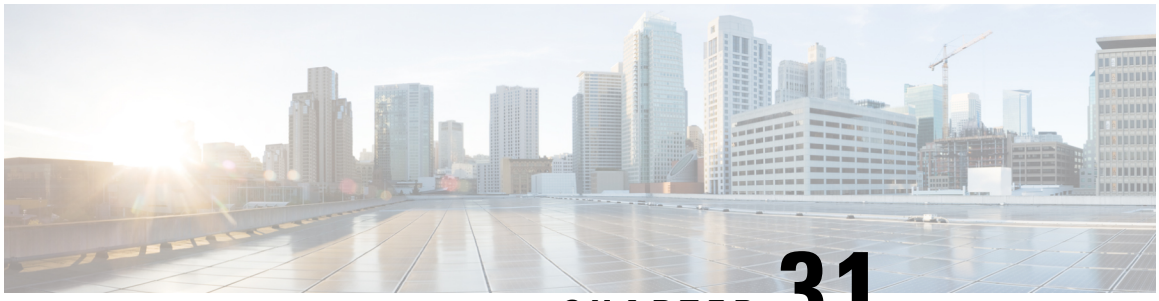
- **Dos-Attacks**: When the number of DoS attacks exceed the set threshold value, the SNMP trap is generated, and the trap is cleared when the number falls below the threshold value within the time interval configured.
- **Drop-Packets**: When the number of packets dropped exceeds the threshold value, the SNMP trap is generated, the trap is cleared when the number falls below the threshold value within the time interval configured.

- **Deny-Rule**: When the number of Deny Rules exceeds the threshold value, the SNMP trap is generated, the trap is cleared when the number falls below the threshold value within the time interval configured.
- **No-Rule**: When the number of No Rules exceeds the threshold value, the SNMP trap is generated, the trap is cleared when the number falls below the threshold value within the time interval configured.

Reassembly Behavior Change

Following are the details about the CUPS reassembly, which are different from the non-CUPS architecture:

- In non-CUPS architecture, with the default FW configuration, fragments are buffered up to 64K bytes. Beyond 64K, all buffered and subsequent fragments are dropped. In non-CUPS architecture, this 64K limit was configurable from 30000 -> 65535. In CUPS, it is possible to reassemble the packet size of maximum 9k in a maximum of six fragments.
- Following are the four CLIs from the non-CUPS architecture that are deprecated in the CUPS:
  - firewall dos-protection teardrop
  - firewall dos-protection ipv6-frag-hdr nested-fragmentation
  - firewall max-ip-packet-size <30000-65535> protocol non-icmp
  - o firewall max-ip-packet-size <30000-65535>protocol icmp
- The following is a single CLI that covers teardrop attack, nested fragmentation, and general ip-reassembly-failure. Max-ip-packet size support is limited to six fragments (~9000 bytes).
  - o Firewall ip-reassembly-failure
- Following are the counters in firewall statistics, that gets incremented for all the attacks related to reassembly.
  - Packets Dropped due to IPv4 Reassembly Failure
  - Downlink Dropped Bytes on IPv4 Reassembly Failure
  - Uplink Dropped Bytes on IPv4 Reassembly Failure
  - Packets Dropped due to IPv6 Reassembly Failure
  - Downlink Dropped Bytes on IPv6 Reassembly Failure
  - Uplink Dropped Bytes on IPv6 Reassembly Failure



CHAPTER 31

GTPC Peer Record and Statistic Optimization

- Revision History, on page 251
- Feature Description, on page 251
- How it Works, on page 251
- Limitations and Restrictions, on page 252
- Configuring the Peer Salvation Functionality, on page 252
- Monitoring and Troubleshooting, on page 254

Revision History

Revision Details	Release
First introduced	21.26

Feature Description

When the Gateway receives the first GTPC message from a peer, the new peer record entry is added to the Session Manager and Demux. This new peer record entry is also propagated to all Session Managers. This process occurs even if a particular GTPC peer does not have any active sessions. This causes accumulation of inactive peer records objects, which results in excess memory usage of the Session Manager and Demux, thereby causing memory overrun of affected proclets. To address this limitation, a new keyword, **peer-salvation** has been added to the existing **gtpc** CLI in the Context Configuration mode.

How it Works

When the peer-salvation keyword is enabled at the context level, it supports the following behavior:

- When a peer goes inactive with zero number of active sessions, the timestamp is stored at that peer record object and a peer record is inserted into the inactive peer list.
- If any new session gets added to inactive peer, the timestamp is reset to zero and the peer record entry is removed from the inactive peer list to avoid salvation of the reactivated peer.

- A one-hour timeout is set per egtpmgr instance level that gets disabled when the keyword is disabled at the context level.
- Separate salvation timers run for egtpinmgr and egtpegmgr.
- By default, (when keyword is not enabled) salvation timer does not run to minimize the memory and CPU impact.

Demux Session Recover Scenario

When Demux proclet crashes or restarts, all the information related to all the inactive peers is cleared in the proclet and is not added again during the session recovery of Demux. These inactive peer records accumulated on the Demux-serving Session Managers might not get salvaged. The peer salvation functionality reconstructs the inactive peer list a the recovered Demux. The last activity timeout for the inactive peers is set to the timestamp of Demux recovery, thereby, allowing Demux to work even after Demux recovery.

Demux Inter-Chassis Session Recovery Scenario

The peer-salvation keyword can be configured on the Active and Standby chassis. When configured, it can even salvage the inactive peers accumulated on the Standby chassis.

Session Manager Session Recovery/ICSR Scenario

Configuring the peer-salvation keyword does not impact the Session Manager recovery or ICSR and vice versa.

Limitations and Restrictions

Following are the known limitations and restrictions while enabling the peer-salvation keyword:

- When the peer-salvation keyword is enabled at the context level, but not enabled at egtp-service level, then peer salvation does not occur.
- All the information (peer statistics/recovery counter and so on) of the particular peer is lost after it is salvaged.
- The context level configuration is applied to egtpinmgr and egtpegmgr separately.
- The min-peers value should be applied judiciously to ensure that the Session Manager in a fully loaded chassis does not go into warn/over state with many peer records. If the Session Manager goes into a warn/over state, then it is recommended to configure a lesser value for min-peers to ensure that the peers are salvaged.
- min-peers configuration is not considered during a new peer creation.
- Only peers with zero number of sessions are salvaged for the configured timeout value. Non-zero number of sessions is not salvaged even if there are few.

Configuring the Peer Salvation Functionality

The following section provides the configuration commands to enable or disable the feature.

gtpc peer-salvation (context configuration mode)

Use this command to enable peer salvation for inactive GTPv2 peers for EGTP services in this context. The peer-salvation keyword is introduced in the Context Configuration mode. Minimum peers and timeout values can be provided with this CLI, which is per egtpmgr (separate for egtpinmgr and egtpegmgr) and across all the egtp-services configured in that context.

To configure peer-salvation in the Context Configuration mode, enter the following commands:

```
configure
  context  context_name
    [ no ] gtpc peer-salvation min-peers value timeout value
    end
```

Notes:

- **no**

  : Disables peer salvation at the context level.

- **peer-salvation**

  : Enables peer salvation for inactive GTPv2 peers for EGTP services in this context.

- **min-peers***value*

  : Configures the minimum number of accumulated GTPv2 peers across all EGTP services to start salvaging the inactive peers. The value ranges from 2000 to 12000.

- **timeout***value*

  : Configures the peer salvation timeout. The peer that is inactive for salvation time is salvaged, specified in hours. The value ranges from 1 to 48 hours.

- This command is disabled by default.

gtpc peer-salvation (eGTP service configuration mode)

Use this command to enable peer salvation for inactive GTPv2 peers for EGTP services in this context. The peer-salvation keyword is added to the existing gtpc command in eGTP Service Configuration mode. When enabled, this functionality is enabled at the specific egtp-service level. This functionality should be enabled at the context level if it is enabled at the egtp-service level. The configuration sequence is not dependent on enabling this functionality.

To configure peer-salvation in the eGTP Service Configuration mode, enter the following commands:

```
configure
  context context_name
    egtp-service egtp_service_name
    [ no ] gtpc peer-salvation
    end
```

Notes:

- **no**

  : Disables peer salvation at the context level.

- **peer-salvation**

  : Enables peer salvation for inactive GTPv2 peers for EGTP services in this context.

- This command is disabled by default.

Monitoring and Troubleshooting

This section provides information regarding show commands and/or their outputs in support of this feature.

Show Commands and/or Outputs

The output of the following CLI command has been enhanced in support of the feature.

gtpc peer-salvation debug-mode debug-min-peers *value1* **debug-timeout** *value2*

show egtp-service all

The output of this command has been enhanced to include the following new field in support of the Peer Salvation functionality:

- GTPC Peer Salvation

show session subsystem debug-info

The output of this command has been enhanced to include the following new fields in support of the Peer Salvation functionality:

- Peer Salvation Stats
  - No of peer salvation requests received on sessmgr.
  - No of peer salvaged on sessmgr.

show demux-mgr statistics egtpinmgr all

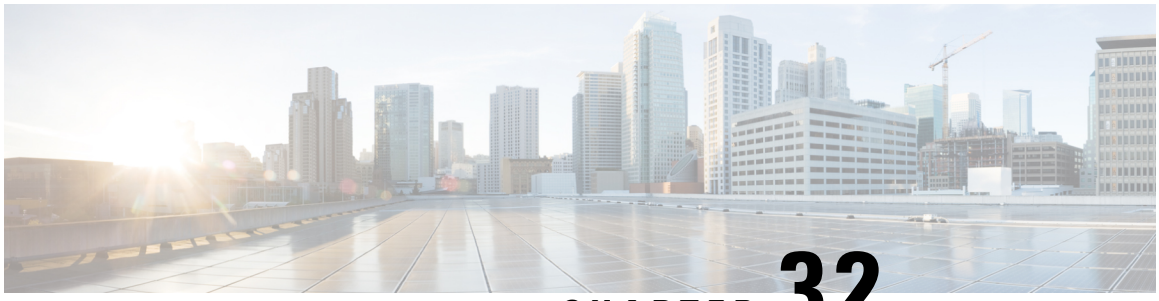
The output of this command has been enhanced to include the following new fields in support of the Peer Salvation functionality:

- Peer Salvation Stats
  - No of peer salvation requests sent by demux.
  - No of peer salvaged on demux.

show demux-mgr statistics egtpegmgr all

The output of this command has been enhanced to include the following new fields in support of the Peer Salvation functionality:

- Peer Salvation Stats
  - No of peer salvation requests sent by demux
  - No of peer salvaged on demux



CHAPTER 32

Gx-alias Enhancement

- Revision History, on page 257
- Feature Description, on page 257
- How it Works, on page 257

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The Gx-alias enhancement feature is a method of installing multiple sets of predefined rules with a single Gx-alias rule name. This rule name comes from PCRF and is transparent to PCEF, where PCRF either activates or deactivates by naming each rule.

This feature is applicable for rules that are installed only on default bearer. To successfully install large number of rules, you must configure **no policy-control update-default-bearer** CLI command under the ACS configuration mode or the **no tft-notify-ue-def-bearer** CLI command under the ACS Rulebase configuration mode to implement it on a per-rulebase level. All the ruledefs, defined under the Gx-alias Group of Ruledef (GoR), must also be defined under the rulebase for it to get applied to the session.

How it Works

The CP expands the GoR for Gx-alias, allocates the PDR IDs to these installed rules, and carries the information in a vendor-specific TLV. As part of this information, the Gx-alias name with Start and End of the PDR IDs are sent to the UP. The UP, after receiving this new TLV, expands the Gx-alias into ruledefs and maps the corresponding PDR IDs in a sequence which is governed by the configuration on UP.

The functionality/behavior of the Gx-alias Enhancement feature includes:

- Before and after the configuration updates, contents of the Gx-alias GoR are exactly the same, and in the same order, on both CP and UP.
- Addition of a new ruledef in a Gx-alias GoR is applied only to new sessions. Only deletion of a ruledef from a Gx-alias GoR is handled in existing session.
- Predefined rules functionality at UP has no impact when Gx-alias is mapped to the ruledefs. That is, URR-IDs/charging is transparent to Gx-alias being used.

NOTE:

- Maximum limit of GoRs that can be configured: 64
- Maximum number of rules allowed per GoR: 512
- Maximum rules allowed per default bearer: 2048

IE Format of Gx-alias

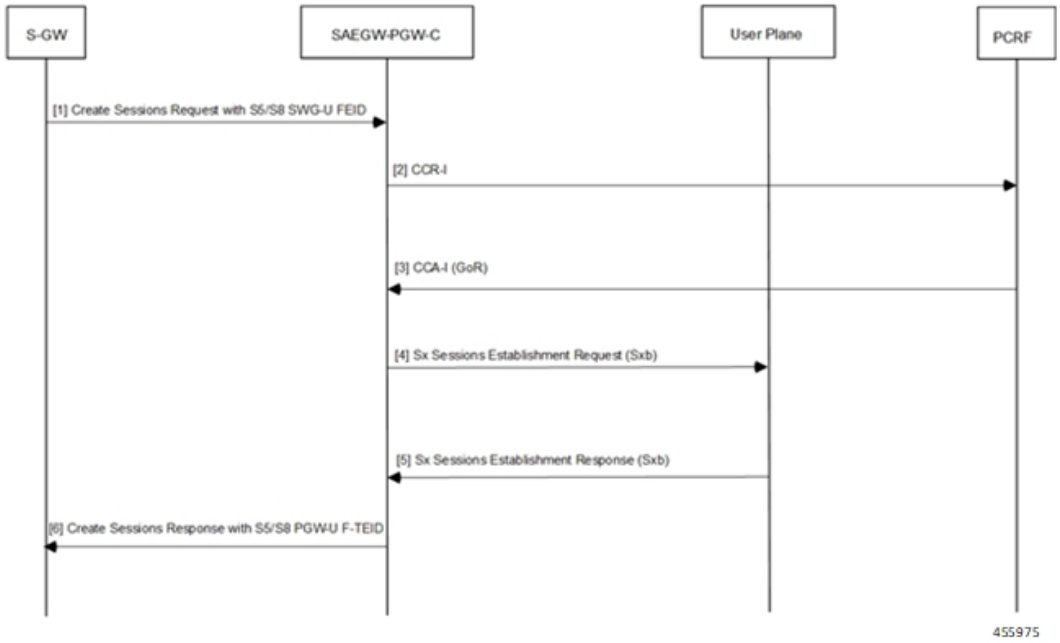
The following table provides the IE Format and encoding information of the Gx-alias feature.

	Bits								
Octets	8	7	6	5	4	3	2	1	
1 to 2	Type = 246 (decimal)								
3 to 4	Length n [Min=7, Max=69 {5+ACSCTRL_GRP_OF_RDEFS_NAMELEN (64)}]								
5	Flags (Add/Delete GoR Rules) For example: 1 for Add, 0 for Delete rules in GoR								
6 to 7	Start PDR ID								
8 to 9	End PDR ID								
10 to n+4	Gx-alias GoR name (min size=2, max size=64)								

PFCP_IE_GX_ALIAS: IE to communicate a Gx-alias GoR name, Start and End PDR IDs, and also the operation to perform from Control Plane to User Plane during Sx Session Establishment/Modification Request message.

Call Flow

This section describes the Gx-alias enhancement call flow.



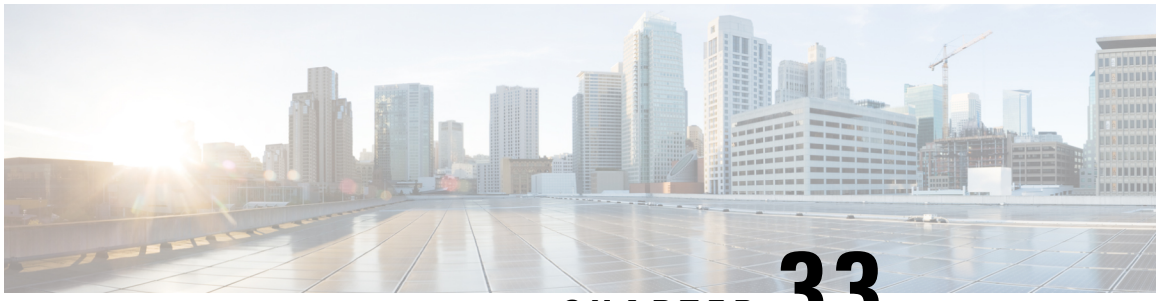
Step	Description
1	S-GW sends a Create Sessions Request with S5/S8 SGW-U FEID to SAEGW-PGW-C.
2	SAEGW performs Gx communication CCR-I with PCRF. During a Pure-P call for CUPS SAEGW, the SAEGW-PGW-C does the following: • After Gx interaction, performs Gx communication (CCR-I and CCA-I) with PCRF. • Performs User Plane selection based on User Plane profile configured with IP pool (APN associated IP pool). • Establishes GTP-U session required for RA/RS for IPv6/IPv4v6 PDN. • Performs Sxb interaction with the selected User Plane.
3	PCRF performs Gx communication CCA-I with SAEGW. Sx Establishment Request session contains the following information: • GoR/GoR Action/FAR/URR information for uplinks and downlink data path: dynamic/predefined/static rules. • Also, Control Plane requests User Plane to allocate F-TEID for P-GW ingress, PDR S5/S8 PGW-U F-TEID. In Gx-alias GoRs, ruledefs must be within the same order for Control Plane and User Plane that are part of Day-0 configuration. The newly configured rules apply only to new sessions that are Cisco-specific Control Plane and User Plane node pairs.

Step	Description
4	SAEGW establishes a Sx Sessions Establishment Request (Sxb) with the User Plane. The new IE format for Gx-alias, PFCP_IE_GX_ALIAS does the following actions: • Communicate a Gx-alias GoR (Group-of-Ruledef) name • Start/End PDR IDs • Perform operations from the Control Plane to the User Plane during the Sx Session Establishment/Modification Request message.
5	The User Plane provides "P-GW ingress PDR S5/S8-U PGW F-TIED" information as part of Sx Session Establishment Response and establishes a Sx Sessions Establishment Response (Sxb) with SAEGW-PGW-C.
6	On receipt of the Sx Session Establishment Response, SAEGW-PGW-C sends Create Session Response towards S-GW with "S5/S8-U PGW F-TEID".

Limitation

Following are the known limitations of the feature:

- IE-handling is applicable only between Cisco-supported Control Plane-User Plane nodes. All ruledefs configured in Gx-alias GoR are bound only to the default bearer.
- To avoid exceeding the recovery time, only eight GoRs are recovered during session recovery. The maximum recommended limit of GoRs to be configured is eight (8).
- With 2048 rules, you may see an impact on scaling of sessions. The maximum recommended rules per default bearer is 1000.



CHAPTER 33

Gy Multiple MSCC and FUI-Redirection

- Revision History, on page 261
- Feature Description, on page 261

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

FUI Redirection is supported in the CUPS architecture. The Final-Unit-Indication AVP can be present in the CCA from the server to indicate that the given quota is the final quota from the server and the corresponding action as specified in the AVP needs to be taken.

If the Final-Unit-Indication AVP is present at MSCC level, and if the Final-Unit-Action AVP is set to TERMINATE, a CCR-Update is sent at the expiry of the allotted quota and report the usage of the category that is terminated.

In the Final-Unit-Indication AVP, if the Final-Unit-Action is REDIRECT or Redirect-Server AVP is present at command level, redirection is performed.

The redirection takes place at the end of consumption of quota of the specified category. The Gy sends a CCR-Update without any RSU or Rating-Group AVP so that the server does not give any more quotas. If the Final-Unit-Action AVP is RESTRICT_ACCESS, then according to the settings in Restriction-Filter-Rule AVP or Filter-Id AVP. Gy sends CCR-Update to the server with used quota.

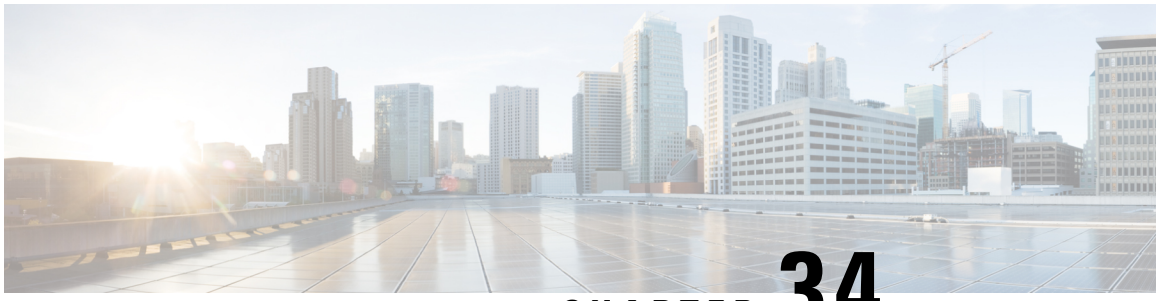
With this release, the following functionalities are supported:

- The FUI-Redirection with HTTP URL.
- The FUI-Redirection is done for the HTTP GET request.

- Only the following CLI: **diameter redirect-validity-timer immediate**, is supported
- As per the 3GPP specification, Redirected-Traffic will also get redirected if the rule is executed from FUI-Redirect. However, provision to allow redirected-traffic to pass through is not present but the CLI behavior with respect to **no diameter fui-redirected-flow allow** behavior is implemented.

Limitations

- The FUI-Redirection along with Filter-Ids/Filter-Rules are not supported.
- Appending the Original URL is not supported.
- Token based mechanism to exit Redirection is not supported.
- The default option traffic-start in the CLI: **diameter redirect-validity-timer**, is not supported.
- The WSP protocol is not supported in CUPS.
- The CLI: **redirect-require-user-agent**, is not supported. Even if the user-agent is not configured, redirection is functional.



CHAPTER 34

ICSR Bulk Statistics

- Revision History, on page 263
- Feature Description, on page 263
- Configuring the ICSR Bulk statistics Schema, on page 263
- Bulk Statistics, on page 264

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

This feature provides the support for ICSR Bulk Statistics schema on the User Plane.

Configuring the ICSR Bulk statistics Schema

Following is the sample configuration to configure the few of the ICSR schema on the User Plane:

```
configure
  bulkstats collection collection_detail
    bulkstats mode mode_name
      sample interval interval_value
      file file_number
        icsr schema icsr_schema format "ICSR:
switchover-number:%switchover-number% switchover-time:%switchover-time%,
 switchover-reason:%switchover-reason%"
        end
```

Show CLIs

Following are the show command CLIs to fetch the ICSR schema bulk statistics data.

- **Show bulk stats data** - displays criteria contained in the statistics gathering scheme for up to four files. See viewing collected bulk statistics data.
- **bulk force gather** – displays the bulkstats data.
- **show bulkstats schemas** - displays the scheme used to gather statistics including collection and transmission statistics. See verifying your configuration.
- **show bulkstats variables** - displays available bulkstat variables (%variable%) by schema type that can be incorporated into a schema format.
- **show configuration bulkstats brief** - displays the bulkstats configuration at a global scope and displays the server configuration. It does not display the schema configuration.

Bulk Statistics

Run the following CLI on User Plane to check the counters available for ICSR schema.

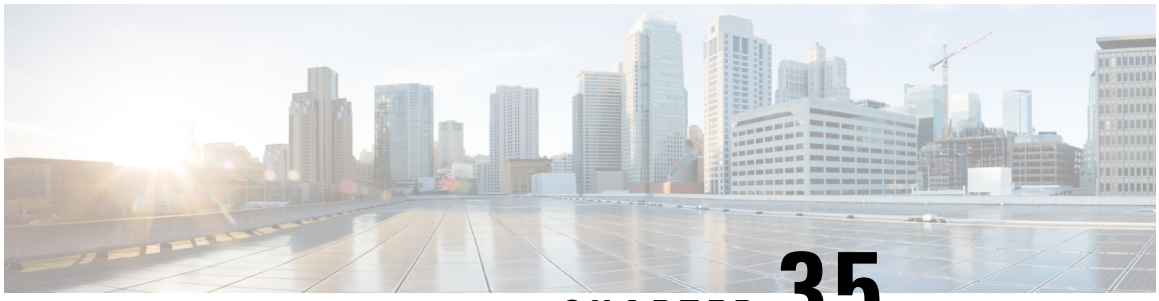
```
show bulkstats variables icsr
```

The following table includes the details of the ICSR counters supported on the User Plane:

Table 11: ICSR Counters applicable in UP

ICSR Counters	Description
switchover-number	Identifying number of switchovers since the last chassis rebooted.
switchover-time	Timestamp for when the switchover was initiated.
switchover-reason	Reason for switchover (manual and BGP failure and auth probe failure and so on).
switchover-duration	Amount of time it took to complete the switchover.
total-num-act-calls-swo-time	Total number of active calls at the time of the switchover.
total-num-lost-calls-swo-time	Total number of data sessions lost due to the switchover.
audit_number	Identifying number of recent audits performed since the last system reboot.
audit_chassis_state	Chassis state (active/standby) on which the audit was performed.
audit_start_time	Timestamp for when the audit was initiated.
ext-audit-sync-start-time	External audit synchronization start time on standby chassis
ready-for-switchover-time	Timestamp on standby chassis when it is ready for next switchover
audit_duration	Amount of time it took to complete the audit
audit_reason	Reason for the audit
total_audit_active_sessions	Total number of active sessions found during the audit.

ICSR Counters	Description
total_audit_new_sessions	Total number of new sessions found during the audit.
total_audit_stale_sessions	Total number of stale sessions found during the audit.
total_audit_inactive_sessions	Total number of inactive sessions found during the audit.
total_sessmgr	Total number of session manager instances on the chassis.
total_sessmgr_active_connected	Total number of session managers in the active-connected state.
total_sessmgr_standby_connected	Total number of session manager instances in the standby-connected state.
total_sessmgr_pending_connected	Total number of sessions manager instances in the pending-connected state.
total_sess_crr_count	Total number of currently existing Call Recovery Records (CRRs).
total_sess_crr_pre_installed	Total number of currently existing pre-installed CRRs.
total-num-act-sessions-swo-time	Total number of fully connected sessions found during the switchover event.
total-num-lost-sessions-swo-time	Total number of fully connected sessions lost during the switchover event
critical-flush-duration	Amount of time it took to complete the critical flush.
total-num-checkpoint-fc-flush	Total number of full checkpoints flushed during switchover
total-num-checkpoint-critical-mc-flush	Total number of critical micro checkpoints flushed during switchover
total-num-checkpoint-mc-flush	Total number of micro checkpoints flushed during switchover
total_first_fc_during_critical_flush	Total number of full checkpoints found during critical flush.
total-num-first-fc-never-sent	Total number of first full checkpoints never sent during switchover
total-num-critical-fc-not-sent	Total number of critical full checkpoints not sent during switchover
checkpoints-never-sent	Total number of SRP checkpoints that were never sent.
checkpoints-send-failed	Total number of sent SRP checkpoints that failed.



CHAPTER 35

Idle Timer for SAE-GW Sessions

- Revision History, on page 267
- Feature Description, on page 267
- Limitations, on page 267
- Configuring Idle Timer for SAE-GW Sessions, on page 268

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

An Idle Timer is supported to identify and remove idle sessions that occur in the SAE-GW.

A session becomes idle in some cases where the session is removed from other network nodes, but due to a technical mishap the session could still remain on the SAE-GW leading to resources being held by these idle sessions.

The Idle Timer, once configured, removes those sessions that remain idle for longer than the configured time limit effectively utilizing the system capacity.

Important This feature is currently restricted to Pure-P and Collapsed Call.

Limitations

The Idle Timer feature does not support recovery of Idle Timer in case of redundancy events.

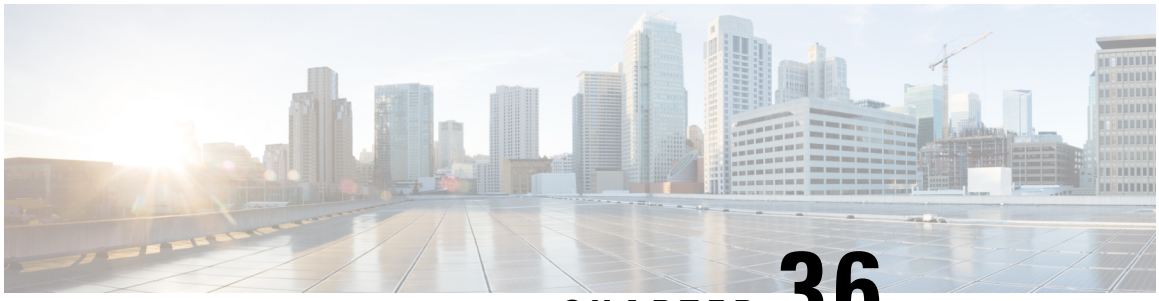
Configuring Idle Timer for SAE-GW Sessions

The Idle Timer is configurable at APN level.

Use the following commands to configure the idle timer for SAE-GW sessions:

```
configure
   context context_name
      apn apn_name
         timeout idle timeout_value
         no timeout idle
         default timeout idle
         end
```

- **no**: Disables the idle timer configuration.
- **default**: Configures the default value for subscriber's time out settings. The default idle timeout value is 0.
- **idle** *timeout_value*: Designates the maximum duration a session can remain idle, in seconds, before system automatically terminates the session. Must be followed by number of seconds between 0 and 4294967295. Zero indicates function is disabled.



CHAPTER 36

IFTASK Hyperthreading

- Revision History, on page 269
- Feature Description, on page 269
- How it Works, on page 269
- Configuring CPU Isolation, on page 270

Revision History

Revision Details	Release
First introduced.	21.25

Feature Description

Hyperthreading uses the Parallel Computing technology to enhance the system performance on processing the packets.

How it Works

IFTASK Hyperthreading ensures that the Poll Mode Driver (PMD)/Multichannel Direct Memory Access (MCDMA) and session manager threads don't coexist on a physical core that is enabled with hyperthreading.

When you enable Hyperthreading, a single core is split into two cores. Hyperthreading ensures that the physical core and its sibling are running the same kind of process, that is, both running either PMD/MCDMA threads or session manager.

The Intel Data Plane Development Kit (DPDK)/IFTASK schedules the PMD and MCDMA threads starting from CPU core number 1 and reserves the core 0 (Master core) for servicing the IFTASK process.

On non-hyperthreaded systems, scheduling PMD and MCDMA threads on any CPU core doesn't have any impact on cache utilization and overall system performance. But, when hyperthreading is enabled, a core and its sibling are scheduled either with PMD/MCDMA or with session manager, resulting in a better system performance. To achieve this, and to maintain the CPU pair, the core numbers are scheduled from 2, instead of starting from 1. The number of IFTASK cores should always be an even number.

CPU Isolation

To enhance the system performance, the CPU which runs the PMD/MCDMA threads is isolated from the kernel. Once the CPU is isolated, the kernel stops scheduling interrupt or other kernel processes.

Limitations and Restrictions

This feature has the following limitations and restrictions in this release:

- This feature is currently supported only on VPC-DI chassis, and not qualified yet for Single Instance (VPC-SI) on Control Plane (CP).
- When you enable or disable "isolcpu", all Service Function (SF) cards are rebooted twice to take effect.
- When you modify IFTASK core configuration with "isolcpu", SF card is rebooted thrice to take effect.
- After enabling hyperthreading, and with single Non-Uniform Memory Access (NUMA) node, the number of PMD/MCDMA cores should be an even number.
- On two NUMA system, the number of PMD/MCDMA cores should be divisible by four, so that the cores are split evenly.

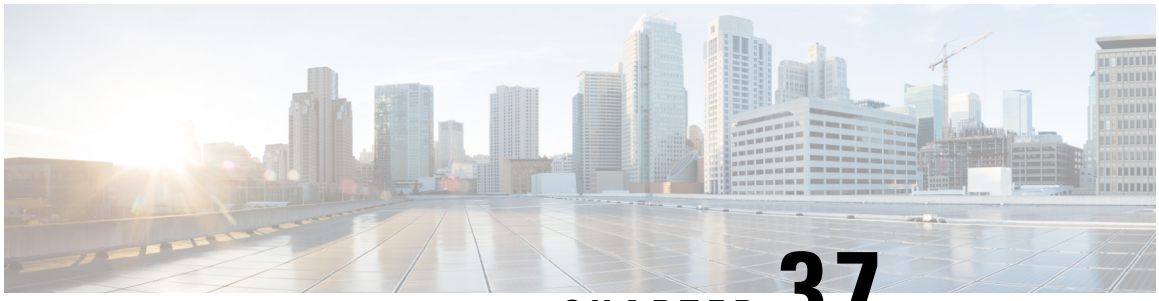
Configuring CPU Isolation

Use the following configuration to enable CPU isolation:

```
config
   iftask
      isolcpu-enable
      end
```

NOTES:

- **iftask isolcpu-enable**: Enables CPU isolation for all SF cards on Virtualized Packet Core - Distributed Instance (VPC-DI) chassis.
- Use the **no iftask isolcpu-enable** CLI command to disable CPU isolation.



CHAPTER 37

Indirect Forwarding Tunnel

- Revision History, on page 271
- Feature Description, on page 271
- How It Works, on page 272
- Configuring Indirect Forwarding Tunnel, on page 275
- Monitoring and Troubleshooting, on page 276

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

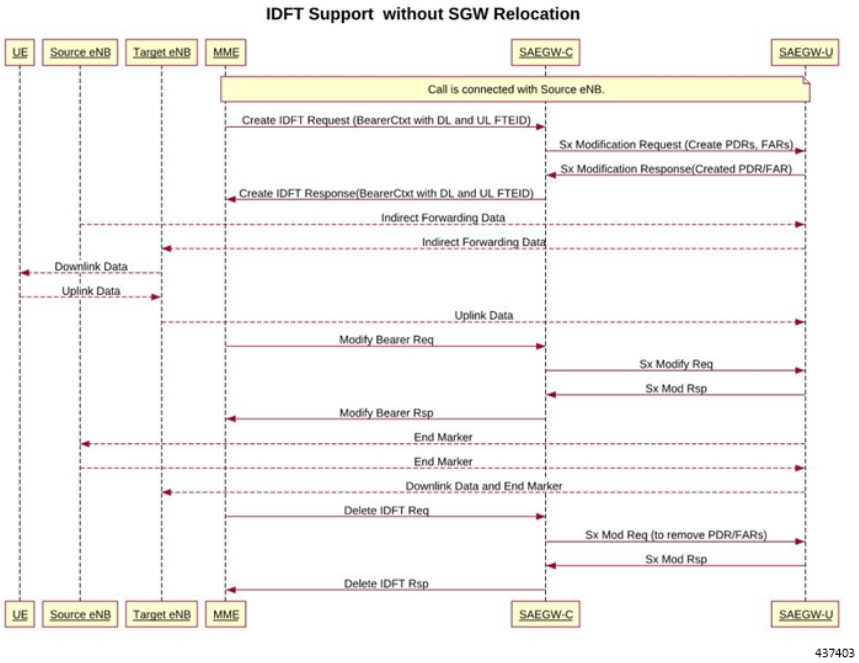
SAEGW supports Indirect Forwarding Tunnel (IDFT) procedures for creation and deletion, which are applicable for Pure-S and Collapsed calls with multi-PDN and multi-bearers. This feature is applicable for IDFT support with and without S-GW relocation and collision scenarios.

Note The IDFT in CUPS is a CLI-controlled feature. By default, the IDFT feature in CUPS is disabled.

How It Works

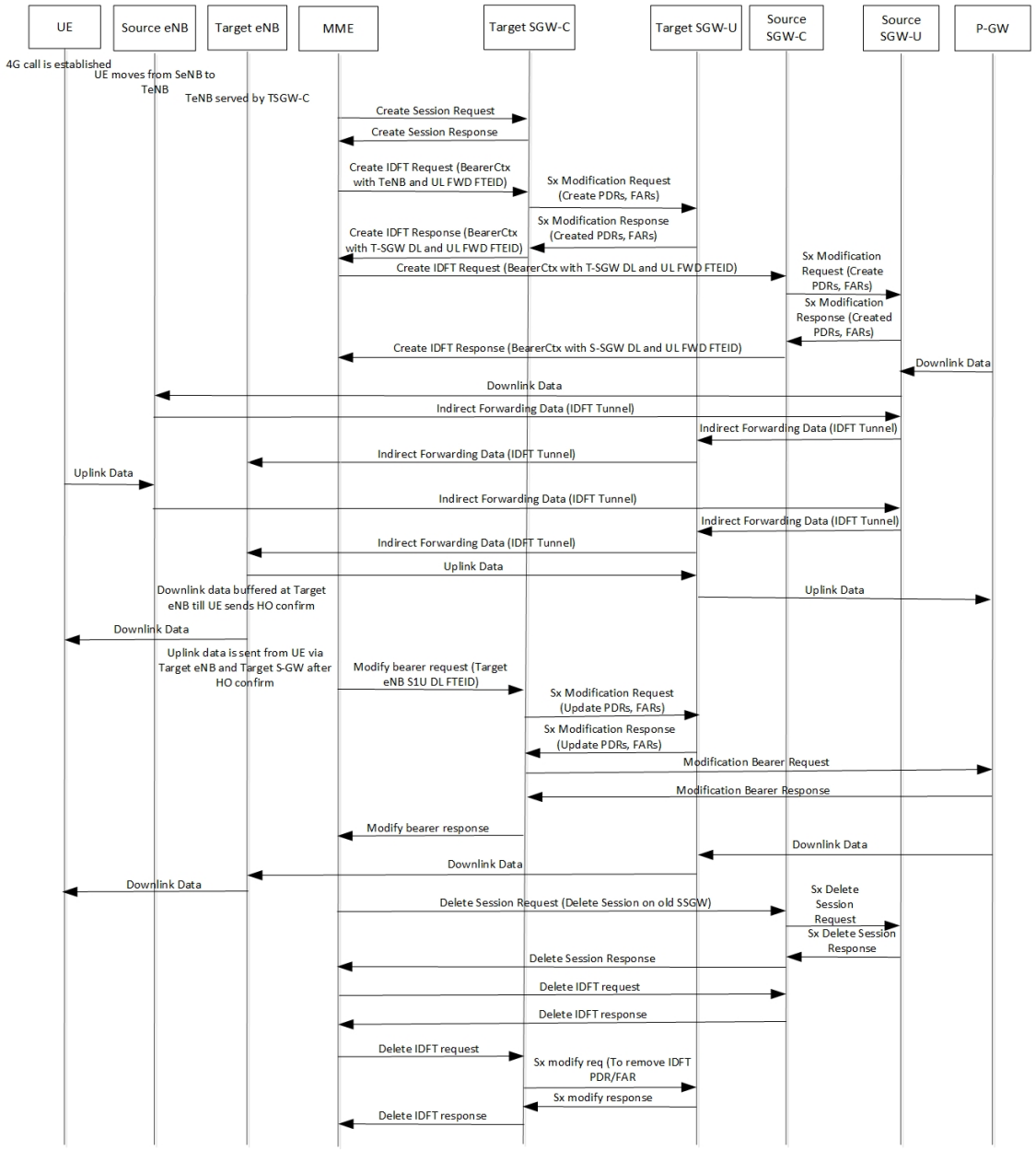
Call Flow

The following call flow illustrates, at a high-level, the IDFT support without S-GW Relocation.



The following call flow illustrates the IDFT support with S-GW Relocation.

Figure 9: IDFT Support with S-GW Relocation



The above call flow describes the IDFT tunnels establishment and deletion with S-GW relocation and without MME change.

If IDFT tunnels are not deleted by MME, then S-GW initiates the local delete of IDFT tunnels.

This feature supports the following scenarios for the Pure-S and Collapse calls:

- S-GW relocation with same MME
- S-GW relocation with same MME and different eNodeB
- S-GW relocation with different MME

- S1-based eNodeB Handoff
- EUTRAN to UTRAN Handoff

Note S4 interface is not supported in CUPS. Hence any EUTRAN to UTRAN handoffs (and vice-versa) involving S4 interface is also not supported.

- EUTRAN to UTRAN Handoff with S-GW relocation
- UTRAN to EUTRAN Handoff
- UTRAN to EUTRAN Handoff with S-GW relocation
- Sx transaction timeout during IDFT setup or removal
- Pending Sx transaction (event from PCRF or OCS) and IDFT request comes in
- Create Bearer Request (CBR) during Active IDFT
- Update Bearer Request (UBR) during Active IDFT
- Delete Bearer Request (DBR) during Active IDFT
- Modify Bearer Request (MBR) behavior on other PDN during Active IDFT
- Source MME path failure
- Target MME path failure
- MME path failure with NTSR enabled
- eGTP-C S5 path failure
- eGTP-C S5 path failure with P-GW restart notification enabled
- Sx path failure (clean IDFT and calls)
- Abort session (clear sub all, local abort, and so on.)
- CBR, UBR on other PDN during Active IDFT
- DBR on other PDN/bearer during Active IDFT
- S1-u path failure for target eNodeB
- S-GW path failure for target S-GW
- S1-u error indication on the default bearer while Active IDFT
- S1-u error indication on the dedicated bearer while Active IDFT
- S1-u error indication from the target S-GW to source S-GW bearer
- S1-u error indication from the target eNodeB to target S-GW bearer

Supported Functionality

The IDFT feature supports the following functionality:

- Create IDFT request for Collapsed, Pure-S, combination of Collapsed and Pure-S multi-PDN calls with multiple bearers.
- Data transfer on downlink and uplink IDFT bearers.
- Deletion of IDFT request from MME. Also, timer-based deletion of IDFT bearer after expiration of a default value of 100 seconds, if the MME does not send an IDFT request for deletion.
- Deletion of IDFT PDN, including Clear/Delete subscribers from MME/P-GW, when normal PDN goes down.
- Sx-Path failure handling in case of Pure-S and collapsed calls during IDFT Active/ IDFT Create Sx-Pending state.
- Message interaction and collision during IDFT PDN establishment or deletion with any other procedure.
- S11/S5 and Sx Path Failure Handling on non-IDFT PDN is now supported when IDFT PDN is Active.

Important Transport GTP-U address capability is assumed to be same across eNodeB and S-GW.

Configuring Indirect Forwarding Tunnel

This section describes the CLI commands available in support of IDFT feature.

Enabling Indirect Forwarding Tunnel Feature

On Control Plane, use the following CLI commands to enable or disable the IDFT feature.

```
configure
  context context_name
    sgw-service service_name
      [ default | no ] egtp idft-support
      end
```

NOTES:

- **idft-support**: Enables/Disables the IDFT feature in CUPS.
- By default, the IDFT feature is disabled and this CLI command is applicable on run-time change.

Verifying the Indirect Forwarding Tunnel Feature

show sgw-service name <service_name>

On Control Plane, the output of this CLI command has been enhanced to display if the IDFT feature is enabled or disabled.

- IDFT-Feature Support for CUPS : Enabled/Disabled

Monitoring and Troubleshooting

This section provides information regarding the CLI commands available in support of monitoring and troubleshooting the feature.

Show Commands Input and/or Outputs

This section provides information regarding show commands and their outputs in support of the feature.

show subscribers saegw-only full all

On Control Plane, use this command to see the IDFT Local and Remote TEID data. The following is a sample output:

```
  Indirect Fwding   : Active
  DL fwd local  addr: 209.165.200.228                DL fwd remote   addr: 209.165.200.226

  DL fwd local  teid: [0x80028004]            DL fwd remote   teid: [0x2002d2e5]
  UL fwd local  addr: 209.165.200.228                UL fwd remote   addr: 209.165.200.226

  UL fwd local  teid: [0x8002a004]            UL fwd remote   teid: [0x20042bca]
```

show subscribers user-plane-only callid <call-id> pdr all

On User Plane, use this command to see the PDR or FAR created for IDFT. The following is a sample output:

Important IDFT PDRs will have ACCESS as the source and destination interface type.

```
+-----Source Interface:          (C) - Core            (A) - Access
|-----Type                       (P) - CP-function     (.) - Unknown
|
|+----Destination Interface:     (C) - Core            (A) - Access
||----Type                       (P) - CP-function     (.) - Unknown
||
||
||+---Rule-Type:                 (S) - Static          (P) - Predefined
|||---Type                       (D) - Dynamic         (.) - Unknown
|||
|||
vvv   PDR-ID    Associated FAR-ID    Associated URR-ID(s)    Associated QER-ID(s)
---   ------    -----------------    --------------------    --------------------
ACS   0x0001         0x8001             n/a                   0x80000001
CAS   0x0002         0x8002             n/a                   0x80000001
```

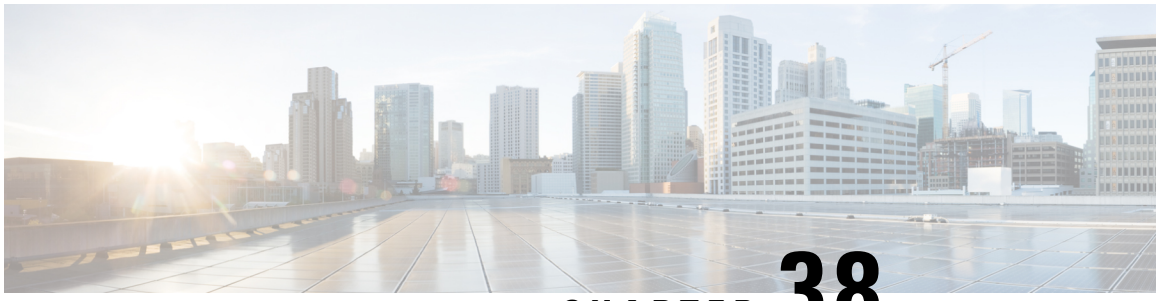
```
ACD   0x0003         0x0003          0x00000007              0x00000002
                                        n/a                 0x80000003
CAD   0x0004         0x0004          0x00000007              0x00000002
                                        n/a                 0x80000003
CAD   0x0005         0x0005          0x00000000              n/a
ACD   0x0006         0x0006          0x00000000              n/a
CAD   0x0007         0x0007          0x00000000              n/a
ACD   0x0008         0x0008          0x00000000              n/a
AAD   0x0009         0x0009          0x00000000              n/a
AAD   0x000A         0x000A          0x00000000              n/a
AAD   0x000B         0x000B          0x00000000              n/a
AAD   0x000C         0x000C          0x00000000              n/a

Total subscribers matching specified criteria: 1
```

show subscribers user-plane-only full all

Important Data statistics on IDFT PDRs are captured in the same way as existing PDR statistics. However, it is captured with a limitation – Statistics for DL and UL IDFT will be incremented in Pkts-Down and Bytes-Down category.

The following is sample output:

```
Static & Predef Rule Match stats:
Rule Name            Pkts-Down  Bytes-Down Pkts-Up    Bytes-Up   Hits       Match-Bypassed
  FP-Down(Pkts/Bytes)  FP-Up(Pkts/Bytes)
-------------------- ---------- ---------- ---------- ---------- ---------- --------------
  -------------------  -----------------
catchall                      0          0          3       1368          3              0
          0/0              0/0

Dynamic Rule Match stats:
PDR Id   Pkts-Down  Bytes-Down Pkts-Up    Bytes-Up   Hits       Match-Bypassed
FP-Down(Pkts/Bytes)    FP-Up(Pkts/Bytes)
-------- ---------- ---------- ---------- ---------- ---------- --------------
-------------------    -----------------
0x0004        2       856          0          0          2              0              0/0
                 0/0
0x000b        2       856          0          0          2              0              0/0
                 0/0
0x000c        2       168          0          0          2              0              0/0
                 0/0
```



CHAPTER 38

IP Pool Management

This chapter includes the following topics:

- Revision History, on page 279
- Feature Description, on page 279
- How It Works, on page 280
- Configuring IP Pool Management, on page 286
- Monitoring and Troubleshooting, on page 289

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
Support of UP selection based on the availability of IP pool chunks.	21.26
With this release, the VPN limitation of 100 UPs per context has been removed.	21.25
First introduced	Pre 21.24

Feature Description

When the IP Pool is unused for a large part, it is not an efficient way of utilizing the resources. The User Plane (UP), which are short of IP resources, can benefit if the unused resources are available to them in a dynamic way.

In the CUPS architecture, there is a centralized Control Plane (CP), large number of UPs, and an automatic and efficient way of managing IP Pool across UPs for the following deployments:

- Co-Located CUPS
- Remote CUPS

How It Works

In CUPS architecture, the PDN/IP context at CP distributes the IP chunk resources among multiple registered UPs in a dynamic way. Following sections describes the overall solution.

Handling UP De-Registration

UP de-registration is triggered in the following scenarios:

- Graceful de-registration from UP—In this scenario, Control-Plane-Group association is removed with User-Plane-Service CLI. The IP addresses are released at sessmgr level on CP.
- UP connection failure from CP—This scenario occurs either because of miss of heartbeat from UP to CP, or because UP restarts and CP is communicated about it. When UP restarts, it implies that the reception of a new Restart-Counter at CP of the specified UP.

After the UP de-registration is triggered, the VPNMGR task on CP validates the identity and address of UP with the information available in the VPNMGR database. In case of mismatch, VPNMGR shows the failure message. In case of match, the validation is successful. On successful validation, VPNMGR takes all the assigned and unassigned chunks from both IPv4 and IPv6 pools from the specified UP.

Whether the UP has some used or all unused IPs, VPNMGR starts a 2-minutes timer before carrying out forceful de-registration of the UP. During forceful de-registration, all IP addresses are deleted from VPNMGR database locally, session entries are removed, and all the chunks are placed to the main address pools at CP.

Hold Timer

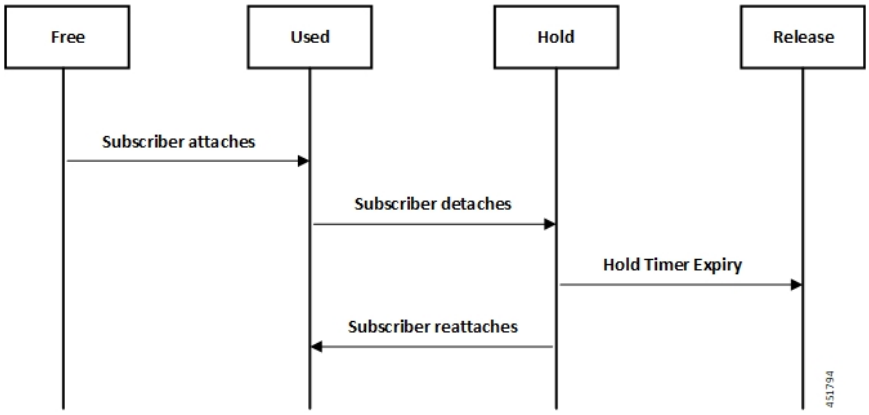
Hold Timer is configured per pool for IPv4 dynamic pools. Static pools and IPv6 pools aren't considered. If Hold Timer isn't configured, an IP address moves from Free to Used state when allocated, and back to Free state when the session is released. With the Hold Timer configured in the pool, a released IP address is moved to Hold state. For the configured Hold Timer duration, the IP address is kept in Hold state and can be reused when the same subscriber attaches again. Since it's in Hold state, the IP address isn't assigned to any other subscriber. After the Hold Timer expiry, the IP address moves to Release state and it's reused when all the free IP addresses are exhausted.

In case of UP deregistration, all IP addresses in Hold state are moved to Free state since the UP details (UP ID and the memory that holds details of UP) aren't preserved at the CP. This might result in the IP address being reused for a different subscriber. Also, VPNMgr recovery and ICSR are supported for Hold addresses.

Address State Change

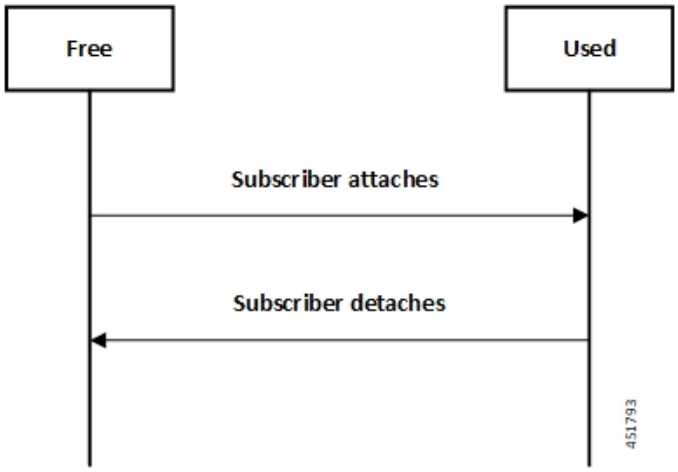
Following call flow describes the address state change with Hold Timer configured.

Figure 10: Address State Change with Hold Timer



Following call flow describes the address state change without Hold Timer configuration.

Figure 11: Address State Change without Hold Timer



Configuring Hold Timer

Use the following configuration to enable Hold Timer feature in CUPS.

```
configure
  context context_name
    ip pool pool_name address-hold-timer seconds
    end
```

NOTES:

- *pool_name*: Specifies the logical name of the IP address pool. *pool_name* must be an alphanumeric string of 1 through 31 characters.
- When the feature is enabled, and an active subscriber is disconnected, the IP address is held or considered still in use, and isn't returned to the free state until the address-hold-timer expires. This enables subscribers who reconnect within the length of time specified (in seconds) to obtain the same IP address from the IP pool. *seconds* is the time in seconds and must be an integer from 0 through 31556926.

Use the **show ip pool address pool-name** *pool_name* CLI command to check the status of all IP addresses in a pool. It also shows the remaining hold time for the held addresses.

IP Pools per Context

You can configure 600 IP pools per UP group in a single context at CP. Also, 2000 IPv4 and 256 IPv6 pools can be configured per context in CP which can be distributed among various UP groups with upper limit of 600 pools per UP. The functionality includes:

- UP group can have a maximum of 600 IP pools for all possible combinations of pool type.
- Pools can be either static, dynamic, or combination of both.
- Pools can be all IPv4, IPv6, or combination of both.
- Out of 600, a UP group can have a maximum of 256 IPv6 pool (context level limitation that is same as ASR5500). All 600 pools can be IPv4.
- If more than 600 IP pools are configured in a UP group, then it can't be determined as to which 600 pool/pool chunks will be allocated to a UP.
- The CP maintains count of routes that are installed at UP. If it exceeds 6000 pool routes (context level limitation that is same as ASR5500), then no new chunk is allocated to UP even if it reaches the threshold for overuse. Similarly, if new IP pool is dynamically allocated and 6000 pool routes are already installed, then no new chunk is allocated from that pool even if pool count is less than 600 for that UP.

As part of this feature, the dynamic IPv4 and IPv6 pool count is replaced with total IPv4 and IPv6 pool count in the **show ip user-plane verbose** CLI command. Also, the output of the CLI command is enhanced to display Total Pool Kernel Routes and Max Pool Kernel Routes fields.

IP Resource Management

In CUPS architecture, the CP has all the IP Pool configurations in PDN/IP context. In compliance with 3GPP standards, the UP registers with CP by Sx Association Request/Response procedure.

During the registration process, the CP finds out all the APNs which are being served by the particular UP, and the associated Pool configuration in each APN. The CP allocates some of the IP chunk resources to a particular UP and sends over the Sx Association Update Request/Response procedure. This information is sent to PDN/IP context instance at UP.

After UP registration is successful, the PDN/IP instance initiates sending of IP chunk resource information to the UP from the Pool. This IP chunk resource information is sent to the UP on Proprietary/Custom IE on Sx Association Update Request/Response message. The PDN/IP instance at UP announces the BGP routes, on per chunk basis, which is received from the CP.

Each UP, which is registered with the CP, is identified using "Peer Id" and the Node ID.

IP Resource Replenishment/Withdrawal Procedure

For efficient utilization of IP resources, the CP allocates IP resources to UP on need basis. And so, it supports replenishment and withdrawal procedures for IP chunk resources.

Based on the threshold logic in CP, it monitors the usage of IP resources in each UP on pool-level basis. If the overall IP chunk usage of the UP crosses certain threshold, the CP sends additional IP chunk resources to the UP.

If certain IP chunks in the UP are not utilised, and idle for certain duration, the CP withdraws those IP chunk resources from respective UPs. For details, see *Configuring Percentage of Chunks Per Pool* section.

No-chunk-pool for One UP per UP Group

Feature Description

For static IP address allocation, the SessMgr requests for specific IP address. The VPNMgr searches for that specific IP address. If the chunk is already allocated to a particular UP, then the VPNMgr allocates that address and responds to the UP which serves the call. For static IPv4v6 call, the requested IPv4 and IPv6 address might belong to different UPs and therefore, success of IPv4v6 can't be guaranteed unless there's only one UP per UP Group. So, for successful static IPv4v6 call, only one UP per UP group can be configured. For one UP per UP Group use case, pool chunking isn't recommended as only one UP uses that pool, and the entire pool can be allocated to the UP rather than in chunks. Also, there are certain use cases to contain one APN to one UP. To support both these use cases, an option to not chunk the pool in CUPS architecture is introduced.

Without the no-chunk-pool functionality, if number of usable addresses are less than the chunk size, then minimum of two chunks were configured.

With no-chunk-pool functionality, a pool can be configured without being chunked. The entire pool is allocated to the UP that is first to request for the pool.

Note The no-chunk-pool functionality is recommended only for a setup with one UP per UP Group. It's not recommended for multi-UP per UP Group.

How it Works

The no-chunk-pool functionality includes:

- When a pool is configured as no-chunk-pool, then pool itself is considered as a chunk and the entire pool is allocated to the UP that is first to request for the pool.
- No-chunk-pool can be public, private, or static.
- No-chunk-pool can be configured within VRF.
- For multi-UP per UP Group, the entire dynamic no-chunk-pool is allocated to the UP that is first to do Sx-association.
- For multi-UP per UP Group, the static no-chunk-pool is allocated in round-robin algorithm among currently servicing UP.
- For multi-UP per UP Group, the dynamically added new pool can get allocated to any UP in UP Group and can’t be deterministically known.

Configuring No-chunk-pool

Use the following configuration to enable no-chunk-pool functionality.

```
configure
   context context_name
      cups enable
         ip pool pool_name ip_address/subnet_mask no-chunk-pool
         ipv6 pool pool_name prefix ip_address/length no-chunk-pool
   exit
```

The no-chunk-pool can be identified from the output of the following CLI commands if the "total-chunks" field displays 1 (one) for that particular pool.

- **show ip pool-chunks pool-all**
- **show ipv6 pool-chunks pool-all**

Static IP Pool Management

In CUPS architecture, the strategy to manage static IP pools differs from dynamic pool managemnent. Static IP pools are broken down into "static-chunks" similar to how dynamic pools are chunked. However, these static chunks are not distributed to the UPs and remain at the CP until a UE requests for the first static address in a certain Static-IP-Chunk during session creation.

The CP selects the UP using the round-robin algorithm and the entire Static-IP-Chunk, to which the requested static address belongs, is assigned to the selected UP. Therefore, whenever any UE requests static addresses (IPv4 or IPv6) from that chunk, the UE is assigned that UP.

Note

- Within dynamic pools, "allow static" is not supported.
- IPv4v6 static PDP is not supported with multiple UPs in a UP Group.
- For the static IPv4v6 PDNs to be successful, both IPv4 and IPv6 addresses must be on the same UP. Only way to ensure this is to have a single UP in the UP group.
- For the multi-PDNs on same APN to be successful, with one PDN as static and the other as dynamic, both addresses must be on same UP. Only way to ensure this is to have a single UP in the UP group.
- In case of static IP pool, address is already decided by UE and so, the benefit of UP selection does not remain.

UP Selection

In CUPS architecture, during the establishment of sessions, UP selection happens among the registered UP. There are various ways to select UP. In earlier releases, Round-Robin Algorithm based UP selection was supported. Currently, least connection User Plane selection algorithm is supported.

UP Selection based on IP Pool Chunk Availability

Prior to 21.26 release, the CP selects an UP based on least session usage or Round-Robin algorithm. If chunks are exhausted in a selected UP, it results in rejection of Session Establishment request by the CP until new IP pools are added for the impacted APNs. This result in wastage of IP resources in an UP, which still has some chunks with free IP addresses.

In 21.26 and later releases, this feature is enhanced to allow UP selection based on the availability of IP pool chunks. When chunks are exhausted in some UPs, and if the CP receives an attach request, the CP selects randomly any UP that has IP addresses available. Also, it ignores any other UP selection algorithm that is configured.

Limitations

- UP selection for Pure-S calls isn't supported.
- Only non-DNS based UP selection is considered for IP address-based validation.
- UP selection is overridden if the selected UP has no IP address to allocate for the session.
- During Sx association, if one of the contexts don't have sufficient chunks for all UPs, then only UPs which get chunks are maintained in VPN.
- As the VPN overrides UP selection when chunks aren't available in certain UPs, you must follow proper IP pool planning guidelines to minimize uneven load distribution across UPs.

  For IP pool planning guidelines, see IP Pool Planning Guidelines, on page 821.

Supported Functionality

The following functionalities are supported as part of the IP Pool Management feature.

- IPv4, IPv6 Public, and private pool-based IP address allocation.
- IPv4 static type address allocation.
- Session Manager recovery and VPN Manager recovery for active calls types.
- CP to CP Interchassis Session Recovery (ICSR) support.
- Hold-timer for IPv4 pools.
- Busy-out (basic functionality) for IPv4 and IPv6 pools.

Limitations

Following are the known limitations and restrictions of this feature for this release:

- The "allow-static" type pool configuration isn't supported.
- Configure the **cups enabled** CLI before you add a pool in IP context to enable IP Pool Management functionality in CUPS mode.
- IPv4v6 static PDP isn't supported with multiple UPs in a UP Group.
- The output of the following CLI commands displays all pools with maximum of 2048 chunks per pool:

- **show ipv6 pool-chunks up-id** *up_id*
- **show ipv6 pool-chunks pool-name** *ipv6_pool_name*
- **show ip pool-chunks up-id** *up_id*
- **show ip pool-chunks pool-name** *ipv4_pool_name*

- Following are not supported in the CUPS architecture:
  - IPv6 – address hold timer is not supported.
  - PDN v4v6 – address hold timer is not supported.
- Upon UE reattach, CP needs to select the same UP session (as IP address is already advertised by that UP in the earlier session). Hence there is no UP load based selection or location based UP Selection possible.
- Hold timer value of 0 is not supported.
- Recovery of Hold timer is supported for up to 1000 address per session manager.
- Reload chassis results in the standby chassis losing the hold timer information.
- Any change to the Hold timer value also requires a Sx re-establishment like it happens for any other pool configuration.

Configuring IP Pool Management

This section provides information about the CLI commands available in support of this feature.

Important

- In an earlier release, User Plane profile configuration was required for S-GW and P-GW. With this release, User Plane profile configuration is no longer required in S-GW and P-GW for UP selection. Also, it is not required to be associated with IP pool configuration.
- Same PDN context should be present at both CP and UP.
- IP context name, which is specified in APN configuration, should be same for both CP and UP.

For guidelines around planning IP Pool and User Plane grouping in your network, contact your Cisco Account representative.

At Control Plane

Enabling IP Context for IP Pool Management

Use the following CLI commands to enable IP context for IP Pool management.

```
configure
   context context_name
```

```
cups enable
end
```

Configuring Custom Threshold Timer

Important In 21.9 (mid-July) and later releases, the **cups chunk-allocate-timer** *allocate_timer_seconds* **chunk-release-timer** *release_timer_seconds* CLI command is deprecated, and replaced by **cups chunk-threshold-timer** *threshold_timer_seconds* and **cups min-chunks-threshold-per-pool** *threshold_percent* CLI commands.

There is a threshold timer for chunk redistribution among UPs. By default, for sending chunk into an over utilized UP, check is carried out every 60 seconds, and for removing chunk from an underutilized UP, check is carried out every 300 seconds. For custom threshold timer, use the following CLI commands:

```
configure
  context context_name
    cups chunk-allocate-timer allocate_timer_seconds chunk-release-timer
release_timer_seconds
    end
```

NOTES:

- This is an optional configuration. If not configured, then by default the allocate threshold is 60 seconds and the release threshold is 300 seconds.
- Use the **default cups chunk-allocate-timer chunk-release-timer** CLI command to revert back the chunk-allocation and chunk-release timer to 60 and 300 respectively.
- If the release timer is configured to be less than the allocate timer, then it is overwritten with the value that equals to the allocate timer.

Configuring Chunk Threshold Timer

Use the following CLI commands to configure CUPS IP pool chunk threshold timer for a context.

```
configure
  context context_name
    cups chunk-threshold-timer threshold_timer_seconds
    end
```

NOTES:

- *threshold_timer_seconds*: Specifies the chunk threshold timer value in seconds, integer 30 to 300. Default = 60 seconds.
- Use the **default cups chunk-threshold-timer** CLI command to set the default value of 60 seconds.
- In releases prior to 21.9 (mid-July), allocation of new chunks to UP and release of chunks from underutilized UP use to occur based on allocation and release timers, respectively. With 21.9 (mid-July) and later releases, only single threshold timer exists, based on which the allocation and release of chunks occur periodically.

Configuring Percentage of Chunks Per Pool

Use the following CLI commands to configure minimum percentage of chunks per pool in a context.

```
configure
   context context_name
      cups min-chunks-threshold-per-pool threshold_percent
      end
```

NOTES:

- *threshold_percent*: Specifies the minimum chunks in percentage of 0 to 50. Default = 10.
- Use the **default cups min-chunks-threshold-per-pool** CLI command to set the default value of 10 percent.
- Chunks are released periodically only when free chunks with particular pools, at CP, are less than the percentage configured with this CLI command.
  - When minimum chunks equals to, or falls below, the configured percentage, a check is done to ascertain if there is any UP that has less than 50% utilization and has more than 2 free chunks. If there is, then one is taken back from each underutilized UP from that particular pool.
- Warning log is generated for: periodicity = chunk-threshold-timer; till minimum chunks in CP VPNmgr are restored.
- UP lockdown period on registration: For first five (5) minutes of a UP registration, no chunks are taken back from that UP and sent to another UP even if other UPs are in need of chunks.

Configuring Chunk-size Value

Use this CLI command to specify the size of the chunk for the particular IP pool during pool creation.

```
configure
   context context_name
      ip pool pool_name prefix mask chunk-size chunk_size_value
      end
```

NOTES:

- Chunk-size configuration happens only during the configuration of IP pool for the first time along with prefix or mask.
- Default Value: 1024

At User Plane

For IP context in UP, there is no requirement for IP Pool configuration, or to use the **cups enabled** CLI command.

Configuring User Planes for a System

Use the following CLI commands to configure maximum number of User Planes expected to be functional in a system.

```
configure
  context context_name
    cups max-user-planes value
    end
```

NOTES:

- In releases prior to 21.25:

  cups max-user-planes *value*: The default value is 10.

  The maximum number of user-planes supported in a context and UP Group is 100.

- In 21.25 and later releases:

  cups max-user-planes *value*: The value is in the range of 1-1000. The default value is 10.

  The maximum number of user-planes supported in a context is increased to 1000.

  This refers to the VPNMGR limits and not the actual number of user-planes that are supported. The actual number of user-planes supported in the system is determined by Sx.

  Use this CLI command to tune the chunks that were initially allocated on Sx-association. It can't be used to restrict the addition of new UPs into a system.

- Use the **default cups max-user-planes** CLI command to revert back the maximum user-planes value to 10.

Monitoring and Troubleshooting

This section provides information regarding monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature at CP.

show ip pool-chunks pool-name <pool-name>

The output of this command displays all the chunks in the specified IPv4 pool.

- chunk-id
- pool-id
- up-id
- total-addr
- free-addr
- used-addr
- hold-addr
- release-addr

- busyout-free
- busyout-used

show ip pool-chunks pool all

The output of this command displays the IPv4 pool chunks that are allocated to all the User Planes.

- chunk-id
- pool-id
- up-id
- total-addr
- free-addr
- used-addr
- hold-addr
- release-addr
- busyout-free
- busyout-used

Note The above fields are also displayed for the **show ipv6 pool-chunks pool all** CLI command except for the "hold-addr" and "release-addr" fields.

show ip pool-chunks up-id <up_id> user-plane-group name <grp-name>

The output of this command displays all the IPv4 chunks that are allocated to a specific User Plane.

- chunk-id
- pool-id
- up-id
- total-addr
- free-addr
- used-addr
- hold-addr
- release-addr
- busyout-free
- busyout-used

show ip user-plane chunks

The output of this command displays IPv4 chunks allocated to each User Plane.

- up-id
- total-chunks
- free-chunks
- used-chunks
- full-chunks

Note The above fields are also displayed for the **show ipv6 user-plane chunks** CLI command.

show ip user-plane prefixes

The output of this command displays IPv4 prefixes allocated to each User Plane.

- up-id
- Total
- Free
- Used
- Hold
- Release
- Busyout-Free
- Busyout-Used

Note The above fields are also displayed for the **show ipv6 user-plane prefixes** CLI command.

show ip user-plane verbose

The output of this command displays all the details related to a User Plane.

- User-plane Group Name
- User-plane ID
- User-plane address
- Sxmgr-id
- IPv4 Chunks
  - Total

    - Free
    - Used
    - Full
- IPv4 address
    - Total
    - Free
    - Used
    - Hold
    - Release
    - Busyout-Free
    - Busyout-Used
- IPv6 Chunks
    - Total
    - Free
    - Used
    - Full
- IPv6 prefixes
    - Total
    - Free
    - Used
    - Busyout-Free
    - Busyout-Used
- Total Pool count
    - IPv4
    - IPv6
- Total Pool Kernel Routes
- Max Pool Kernel Routes
- Total VRFs
- apn-without-pool-name-v4
- apn-without-pool-name-v6
- Pool-groups

show ip user-plane

The output of this command displays the details of all the User Planes that are registered with the VPN Manager.

- up-id
- user-plane-address
- user-plane-group-name
- sxmgr-id

NOTES:

- Use the **show ip user-plane up-id***up_id***user-plane-group name** *grp-name* to view the details of a specific User Plane belonging to a specific User Plane group.

show ipv6 pool-chunks pool-name <pool-name>

The output of this CLI command displays all the chunks in the IPv6 pool.

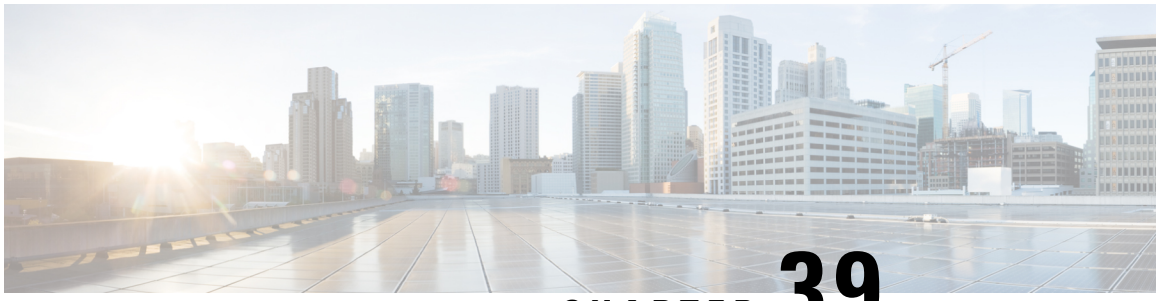
- chunk-id
- pool-id
- up-id
- total-addr
- used-addr
- busyout-free
- busyout-used

show ipv6 pool-chunks up-id <up_id> user-plane-group name <grp-name>

The output of this command displays all the IPv6 chunks that are allocated to a specific User Plane.

- chunk-id
- pool-id
- up-id
- total-addr
- used-addr
- busyout-free
- busyout-used

show ipv6 pool-chunks up-id <up_id> user-plane-group name <grp-name>



CHAPTER 39

IP Source Violation

This chapter includes the following topics:

- Revision History, on page 295
- Feature Description, on page 295
- Configuring IP Source Violation, on page 295
- Monitoring and Troubleshooting, on page 296

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The CUPS architecture supports packet source validation on the User-Plane. Source validation is useful if packet spoofing is suspected or for verifying packet routing and labeling within the network.

The User-Plane checks the source IP address of the uplink data packet with the IP address of the UE for a match and decides to either drop or permit the data packet further based on configured values.

An existing configuration, which is part of the non-CUPS architecture is implemented for this feature. The **ip source-violation** command – part of the *APN Configuration* mode is used to implement packet source validation.

Configuring IP Source Violation

Use the following configuration to enable or disable packet source validation for a given APN:

```
configure
  context context_name
    apn apn_name
      ip source-violation { ignore | check [ drop-limit limit ] [
exclude-from-accounting ] }
      default ip source-violation
      end
```

NOTES:

- **default**: Enables the checking of source addresses received from subscribers for violations, with a drop limit of 10 invalid packets that can be received from a subscriber prior to their session being deleted.
- **ignore**: Disables source address checking for the APN.

  The User Plane does not increment the IP source violation counter and the dropped packet statistics will be zero. The User Plane would create a different Stream, and VPP sends these packets through fastpath using the same Stream ID.
- **check [drop-limit** *limit* **]**: Default: Enabled, limit = 10.

  Enables the checking of source addresses received from subscribers for violations. A drop-limit can be configured to set a limit on the number of invalid packets that can be received from a subscriber prior to their session being deleted.

  limit: can be configured to any integer value between 0 and 1000000. A value of 0 indicates that all invalid packets will be discarded, but the session will never be deleted by the system.
- **exclude-from-accounting**: Excludes the packets identified with IP source violation from the statistics generated for accounting records.

  When **exclude-from-accounting** is disabled:
  - Dropped packets are not accounted. However, the packets that are sent from VPP are charged.
  - Usage Report URR has dropped bytes.
  - Packet drop counter increases.

  When **exclude-from-accounting** is enabled:
  - Dropped packets are not accounted.
  - Usage Report URR will not have dropped packets.
  - Packet drop counter increases.

Monitoring and Troubleshooting

This section provides information regarding monitoring and troubleshooting the IP Source Violation feature.

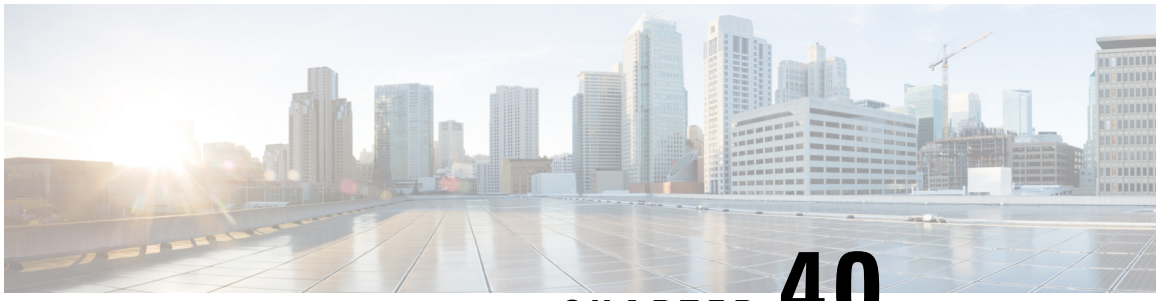
Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show sub user-plane-only full all

On executing the above command, the following fields are displayed for this feature:

- ip source violations



CHAPTER 40

IPSec in CUPS

- Revision History, on page 299
- Feature Description, on page 299
- Limitations and Restrictions, on page 305
- Configuring DSCP in Crypto Map, on page 306
- Configuring QoS, on page 307
- Monitoring and Troubleshooting, on page 308

Revision History

Revision Details	Release
First introduced.	21.25

Feature Description

IPSec is a suite of protocols that interact with one another to provide secure private communications across IP networks. These protocols allow the system to establish and maintain secure tunnels with peer security gateways. IPSec provides confidentiality, data integrity, access control, and data source authentication to IP datagrams.

IPSec AH and ESP

Authentication Header (AH) and Encapsulating Security Payload (ESP) are the two main wire-level protocols used by IPSec. They authenticate (AH) and encrypt-plus-authenticate (ESP) the data flowing over that connection.

- AH is used to authenticate – but not encrypt – IP traffic. Authentication is performed by computing a cryptographic hash-based message authentication code over nearly all the fields of the IP packet (excluding those which might be modified in transit, such as TTL or the header checksum), and stores this in a newly added AH header that is sent to the other end. This AH header is injected between the original IP header and the payload.

- ESP provides encryption and optional authentication. It includes header and trailer fields to support the encryption and optional authentication. Encryption for the IP payload is supported in transport mode and for the entire packet in the tunnel mode. Authentication applies to the ESP header and the encrypted data.

IPSec Transport and Tunnel Mode

Transport Mode provides a secure connection between two endpoints as it encapsulates IP payload, while Tunnel Mode encapsulates the entire IP packet to provide a virtual "secure hop" between two gateways.

Tunnel Mode forms the more familiar VPN functionality, where entire IP packets are encapsulated inside another and delivered to the destination. It encapsulates the full IP header and the payload.

Note The UP:UP ICSR over IPSec works only with Tunnel Mode. Transport Mode isn't supported.

IPSec Terminology

Crypto Access Control List

Access Control Lists define rules, usually permissions, for handling subscriber data packets that meet certain criteria. Crypto ACLs, however, define the criteria that must be met for a subscriber data packet to be routed over an IPSec tunnel.

Unlike other ACLs that are applied to interfaces, contexts, or one or more subscribers, crypto ACLs are matched with crypto maps. In addition, crypto ACLs contain only a single rule while other ACL types can consist of multiple rules.

Prior to routing, the system examines the properties of each subscriber data packet. If the packet properties match the criteria specified in the crypto ACL, the system initiates the IPSec policy dictated by the crypto map.

Transform Set

Transform Sets are used to define IPSec security associations (SAs). IPSec SAs specify the IPSec protocols to use to protect packets.

Transform sets are used during Phase 2 of IPSec establishment. In this phase, the system and a peer security gateway negotiate one or more transform sets (IPSec SAs) containing the rules for protecting packets. This negotiation ensures that both peers can properly protect and process the packets.

ISAKMP Policy

Internet Security Association Key Management Protocol (ISAKMP) policies are used to define Internet Key Exchange (IKE) SAs. The IKE SAs dictate the shared security parameters (such as which encryption parameters to use, how to authenticate the remote peer, etc.) between the system and a peer security gateway.

During Phase 1 of IPSec establishment, the system and a peer security gateway negotiate IKE SAs. These SAs are used to protect subsequent communications between the peers including the IPSec SA negotiation process.

Crypto Map

Crypto Maps define the tunnel policies that determine how IPSec is implemented for subscriber data packets.

There are several types of crypto maps supported in CUPS. They are:

- Manual crypto maps
- IKEv2 crypto maps
- Dynamic crypto maps

Crypto Template

A Crypto Template configures an IKEv2 IPSec policy. It includes most of the IPSec parameters and IKEv2 dynamic parameters for cryptographic and authentication algorithms. A security gateway service won't function without a configured crypto template.

Only one crypto template can be configured per service. However, a single StarOS instance can run multiple instances of the same service with each associated with that crypto template.

DSCP Marking of ESP Packets

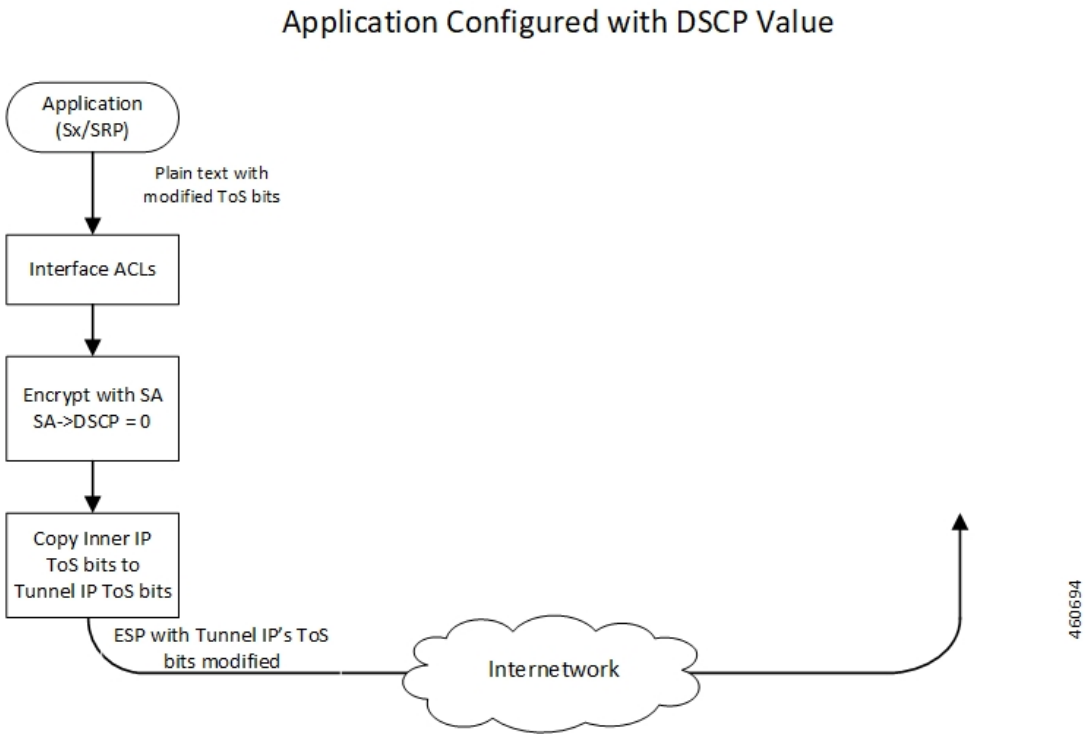
Applications such as SRP, SX, RCM, LI, and TACACS operate between nodes that are deployed across different networks. All these applications require quick turnaround while communicating with remote systems. Marking of Encapsulating Security Payload (ESP) packets with a Quality of Service (QoS) such as Differentiated Services Code Point (DSCP) helps to determine the traffic classification for these types of packets. This feature enables prioritization of IPsec packets within their IP core network, and improves scalability of interfaces such as Sx, and SRP using IPsec.

There are two ways to apply DSCP value on the ESP packets:

- Through Application configured with DSCP value
- Through Crypto Map configured with DSCP value

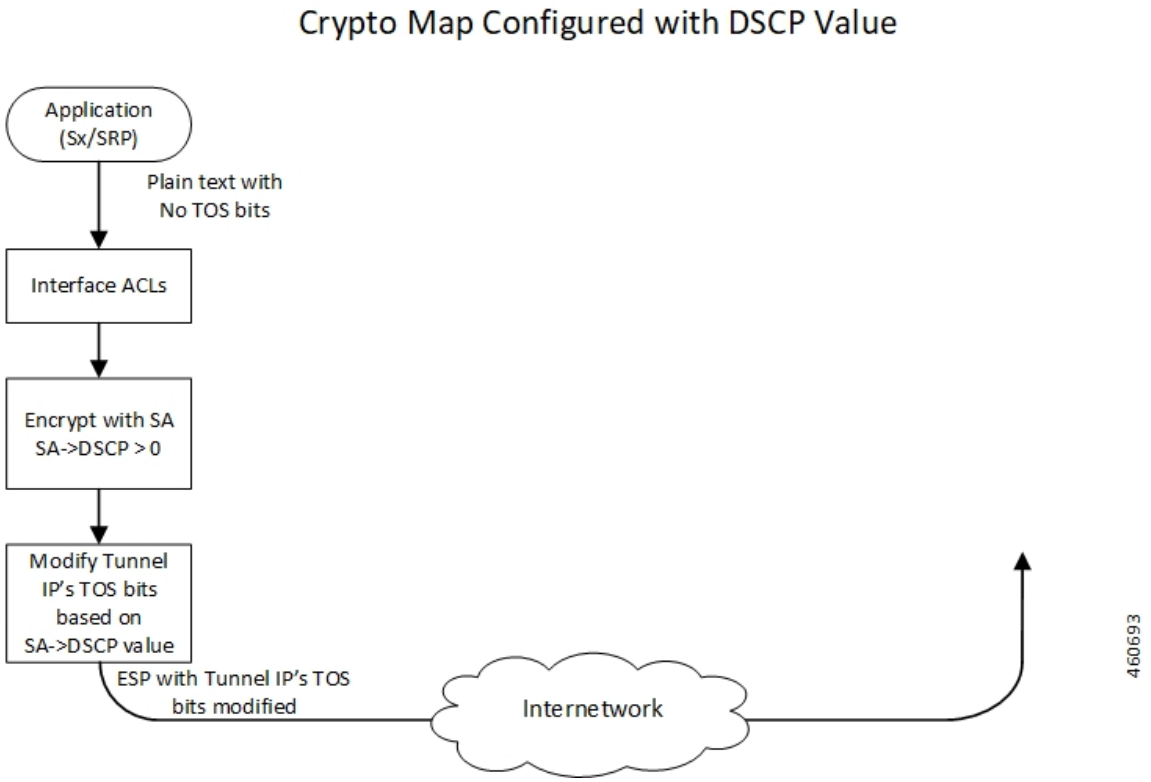
Application Configured with DSCP Value

If an application such as SRP, SX, or LI supports DSCP configuration, the ESP packets after encryption check if the Type of Service (ToS) bits are set in the application IP header. If the ToS bits of the application IP header are non-zero, it copies the inner ToS bits to the ToS bits of tunnel IP header, and egress the packet. The following figure illustrates the operating procedure.



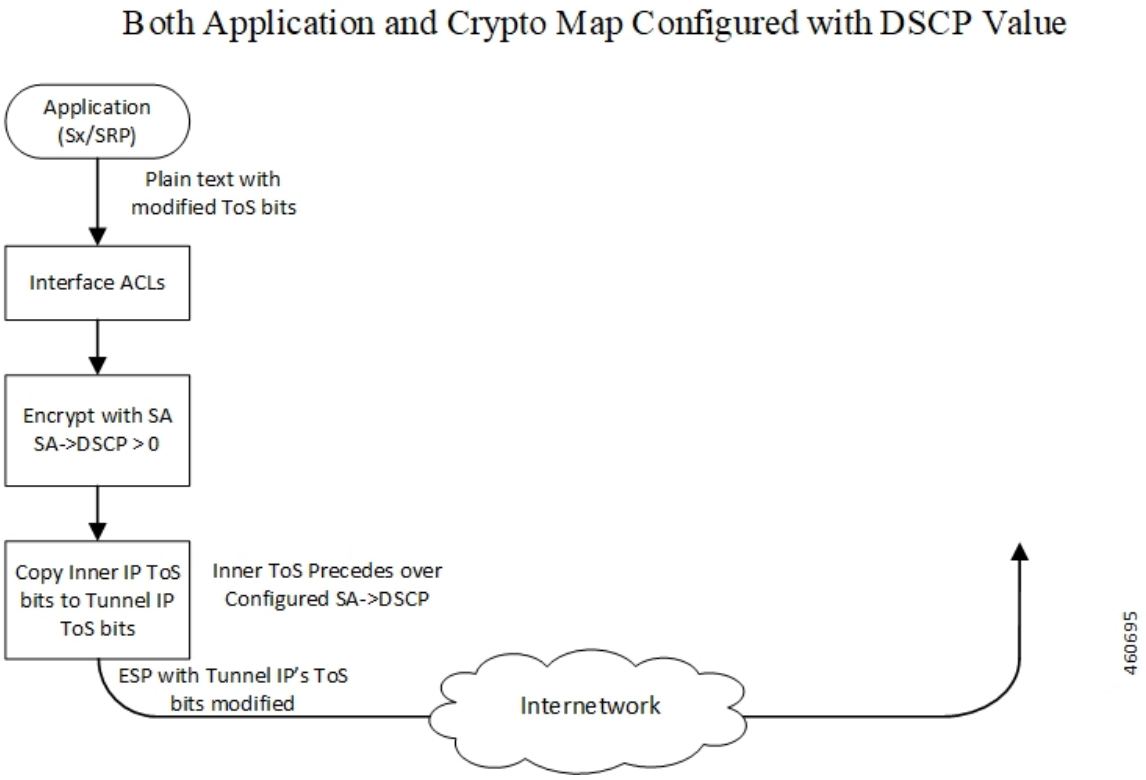
Crypto Map Configured with DSCP Value

Every application that needs to be encrypted has an associated crypto map, which is user configurable. Once the crypto map is enabled on the specific interface, Security Association (SA) database for this crypto map is updated with a DSCP value. A new field is defined in the SA database structure to hold the DSCP value. Once the packet is encrypted, it checks if the SA database has a valid DSCP value. If a valid DSCP value is found, then this DSCP value is copied to the ToS bits of tunnel IP header, and the packet is egressed. The following figure illustrates the operating procedure.



Application and Crypto Map Configured with DSCP Value

If the DSCP value is configured in both crypto map and application IP header, the application ToS bits take precedence, and this value is copied to the ToS bits of Tunnel IP header. The following figure illustrates the operating procedure.



Supported Algorithms

IPSec in CUPS supports the protocols in the table below, which are specified in RFC 5996.

Protocol	Type	Supported Options (without VPP)	Supported Options (with VPP)
Internet Key	IKEv2 Encryption	DES-CBC, 3DES-CBC, AES-CBC-128, AES-CBC-256	
Exchange version 2	IKEv2 Pseudo Random Function	PRF-HMAC-SHA1, PRF-HMAC-MD5, AES-XCBC-PRF-128	PRF-HMAC-SHA1, PRF-HMAC-MD5, AES-XCBC-PRF-128
	IKEv2 Integrity	HMAC-SHA1-96, HMAC-SHA2-256-128, HMAC-SHA2-384-192. HMAC-SHA2-512-256, HMAC-MD5-96, AES-XCBC-96	HMAC-SHA1-96, HMAC-SHA2-256-128, HMAC-SHA2-384-192. HMAC-SHA2-512-256, HMAC-MD5-96, AES-XCBC-96
	IKEv2 Diffie-Hellman Group	Group 1 (768-bit), Group 2 (1024-bit), Group 5 (1536-bit), Group 14 (2048-bit)	Group 1 (768-bit), Group 2 (1024-bit), Group 5 (1536-bit), Group 14 (2048-bit)

Protocol	Type	Supported Options (without VPP)	Supported Options (with VPP)
IP Security	IPSec Encapsulating Security Payload Encryption	NULL, DES-CBC, 3DES-CBC, AES-CBC-128, AES-CBC-256, AES-128-GCM-128, AES-128-GCM-64, AES-128-GCM-96, AES-256-GCM-128, AES-256-GCM-64, AES-256-GCM-96	NULL, DES-CBC, 3DES-CBC, AES-CBC-192, AES-CBC-128, AES-CBC-256, AES-128-GCM-128, AES-128-GCM-64, AES-128-GCM-96, AES-192-GCM, AES-256-GCM-128, AES-256-GCM-64, AES-256-GCM-96
	Extended Sequence Number	Value of 0 or off is supported (ESN itself is not supported)	Value of 0 or off is supported (ESN itself is not supported)
	IPSec Integrity	NULL, HMAC-SHA1-96, HMAC-MD5-96, AES-XCBC-96, HMAC-SHA2-256-128, HMAC-SHA2-384-192, HMAC-SHA2-512-256 **Important** HMAC-SHA2-384-192 and HMAC-SHA2-512-256 are not supported on VPC-DI and VPC-SI platforms if the hardware doesn't have crypto hardware.	NULL, HMAC-SHA1-96, HMAC-MD5-96, HMAC-SHA2-256-128, HMAC-SHA2-384-192, HMAC-SHA2-512-256 **Important** HMAC-SHA2-384-192 and HMAC-SHA2-512-256 are not supported on VPC-DI and VPC-SI platforms if the hardware doesn't have crypto hardware.

Note For more information about IPSec, refer the StarOS *IPSec Reference*. Note that not all features/functionality are applicable for CUPS.

For detailed information about IPSec for Sx, LI, SRP, and so on, refer the relevant chapters in the CUPS CP Guide, CUPS UP Guide, Sx Interface Guide, and CUPS LI Guide.

Limitations and Restrictions

Following are the limitations and restrictions for this feature:

- The feature doesn't support modification of application ToS.
- DSCP value configuration in the crypto map CLI command must be added in the same context where the application is configured as **Day-1** configuration on the UP.
- If the DSCP configuration is applied after the tunnel is created, the associated crypto maps must be re-applied on the interfaces.

- If reordering of packets occurs in an SA, the receiver might discard packets because of anti-replay mechanism.

Configuring DSCP in Crypto Map

Use the following CLI commands to apply the DSCP value for the specific transform set.

```
configure
    context context_name
      ipsec transform-set set_name
        dscp dscp_value
        exit
      exit
    end
```

Sample Configuration

The following is a sample configuration:

```
context ipsec-d
    ip access-list foo0
      permit tcp 209.165.200.225 209.165.200.250 209.165.200.245 209.165.200.250

    #exit

    ip access-list foo1
      permit tcp 209.165.200.225 209.165.200.250 209.165.200.247 209.165.200.250
    #exit
    ipsec transform-set A-foo
    dscp 0x28
    #exit
    ikev2-ikesa transform-set ikesa-foo
    #exit
crypto map foo0 ikev2-ipv4
      match address foo0
      authentication local pre-shared-key encrypted key encrypted_key
      authentication remote pre-shared-key encrypted key encrypted_key
      ikev2-ikesa max-retransmission 3

      ikev2-ikesa retransmission-timeout 15000

      ikev2-ikesa setup-timer 60

      ikev2-ikesa transform-set list ikesa-foo

      ikev2-ikesa rekey

      payload foo-sa0 match ipv4
        ipsec transform-set list A-foo
        lifetime 9000
        rekey keepalive
      #exit
      peer 209.165.201.1

      ikev2-ikesa policy error-notification
```

```
#exit
crypto map foo1 ikev2-ipv4

  match address foo1

  authentication local pre-shared-key encrypted key encrypted_key
  authentication remote pre-shared-key encrypted key encrypted_key
  ikev2-ikesa max-retransmission 3

  ikev2-ikesa retransmission-timeout 15000

  ikev2-ikesa transform-set list ikesa-foo

  ikev2-ikesa rekey

  payload foo-sa0 match ipv4

    ipsec transform-set list A-foo

    lifetime 9000

    rekey keepalive

  #exit

  peer 209.165.201.2

  ikev2-ikesa policy error-notification

#exit
```

Configuring QoS

The ESP packets that are marked with DSCP follow the underlying L2 marking infrastructure.

The configuration to set up QoS based on the DSCP triggers the L2 marking of the ESP packets before egress from the chassis.

The following is a sample configuration:

```
Config
  qos ip-dscp-iphb-mapping dscp 0x28 internal-priority cos 0x1
  qos l2-mapping-table name l2Marktable
     internal-priority cos 0x1 color 0x0 802.1p-value 0x4 mpls-tc 0x6
   exit
end
```

NOTES:

- **qos ip-dscp-iphb-mapping**: Creates a QOS profile.
- **dscp** *dscp_value*: Maps the IP DSCP values to the internal QoS.
- **internal-priority cos** *class_of_service_value* **color** *color_value* **802.1p-value** *mpls_tc_value*: Maps internal QoS priority with COS values.

The following is a sample configuration to associate L2 mapping table in IPsec context:

```
config
    context ipsec-s
```

```
      associate l2-mapping-table name l2Marktable
end
```

NOTES:

- **associate l2-mapping-table**: Maps QoS from internal QoS to l2 values.
- **name** *table_name*: Specifies the name of table to map QoS from internal QoS to l2 values. *table_name* must be an alphanumeric string of size 1–80.

Monitoring and Troubleshooting

This section describes the CLI commands available to monitor and/or troubleshoot the DSCP Marking of ESP Packets feature.

Show Commands and Outputs

This section provides information regarding show commands and their outputs in support of this feature.

show crypto map tag *tag_name*: Use this command to display the configured DSCP value.

```
Map Name: foo0
=======================================

  IPSec Manager: 54
  Map status: Complete
  Payload:
 ACLs:
   foo0
 Rules:
   permit tcp 209.165.200.225 209.165.200.250 209.165.200.245 209.165.200.250 eq 6002
  Crypto Map Type: IPSEC IKEv2 over IPv4
  IKE SA Transform 1/1
    Transform Set:
      Encryption Cipher: aes-cbc-128
      Encryption Accel: None
      Pseudo Random Function: sha1
      Hashed Message Authentication Code: sha1-96
      HMAC Accel: None
      Diffie-Hellman Group: 2
  IKE SA DSCP Value:  0x28

  IKE SA IDi [Peer]: Disabled

  IKE SA DH Exponentials reuse groups : None

  IKEv2 IKESA DDOS Mitigation Params:
    Half Open Timer: Disabled
    Decrypt Fail Count: Disabled
    Max IKEv2 requests Allowed : Disabled
    Message Queue Size: Disabled
    Rekey Rate: Disabled
    Max Certificate Size: Disabled

  IKEv2 Notify Payload:
    Device Identity: Enabled[Default]
  Notify Payload Error Message Type:
    UE: 0
```

```
  Network Transient Minor: 0
  Network Transient Major: 0
  Network Permanent: 0

Blacklist/Whitelist : None

OCSP Status         : Disabled
OCSP Nonce Status  : Enabled
OCSP Responder Address :None
OCSP HTTP version : 1.0


Remote-secret-list: <not-configured>

Authentication Local:
 Phase 1 - Pre-Shared Key (Size = 7)

Authentication Remote:
 Phase 1 - Pre-Shared Key (Size = 7)


Self-Certificate Validation: Disabled
Certificate Server Timeout: 20 Sec
Minimum Certificate Key Size Validation: Disabled


Max Dhost Connections: 40

IPSec SA Payload 1/1
  Name : foo-sa0
  Payload Maximum Child SA: 1 [Default]
  Payload Ignore Ikesa Rekey: Disabled
  Payload Lifetime Params:
    Seconds: 90
    Sequence Number: 4293918720 [Default]
  Payload TSI Start Address: Address Endpoint
  Payload TSI End Address: Address Endpoint

  IPSec SA Transform 1/1
    Transform Set:
      Protocol: esp
      Encryption Cipher: aes-cbc-128
      Encryption Accel: None
      Hashed Message Authentication Code: sha1-96
      HMAC Accel: None
      Diffie-Hellman Group: none
      ESN: Disabled
      Dscp: 0x28
Dont Fragment: Copy bit from inner header
IPv4 Payload fragment type: outer
MTU: 1438 [Default]

NATT: Disabled


IKEv2 Fragmentation: Enabled
IKEv2 MTU Size IPv4/IPv6: 1384/1364

CERT Enc Type URL Allowed: Disabled
Custom FQDN Allowed: Disabled
DNS Handling: Normal [Default]

interface using this crypto-map: saegw-li1-loopback-ipv4
```

```
  Local Gateway: 209.165.202.129
  Remote Gateway: 209.165.201.1
```

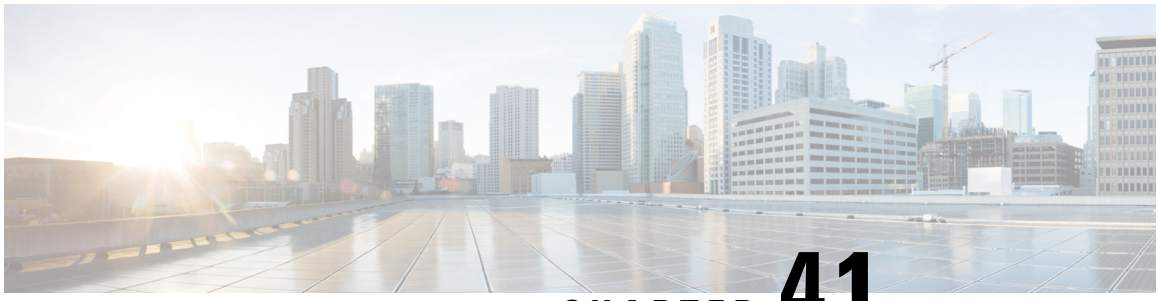
show qos ip-dscp-iphb-mapping: Use this command to display mapping QoS information in a packet to internal-qos marking.

```
--------------+-----------------
    DSCP      |     Internal Qos
--------------+-----------------
   0x00       |          0
   0x01       |          0
   0x02       |          0
   0x03       |          0
   0x04       |          0
   0x05       |          0
   0x06       |          0
   0x07       |          0
--------------+-----------------
   0x08       |          0
   0x09       |          0
   0x0a       |          0
   0x0b       |          0
   0x0c       |          0
   0x0d       |          0
   0x0e       |          0
   0x0f       |          0
--------------+-----------------
   0x10       |          0
   0x11       |          0
   0x12       |          0
   0x13       |          0
   0x14       |          0
   0x15       |          0
   0x16       |          0
   0x17       |          0
--------------+-----------------
   0x18       |          0
   0x19       |          0
   0x1a       |          0
   0x1b       |          0
   0x1c       |          0
   0x1d       |          0
   0x1e       |          0
   0x1f       |          0
--------------+-----------------
   0x20       |          0
   0x21       |          0
   0x22       |          0
   0x23       |          0
   0x24       |          0
   0x25       |          0
   0x26       |          0
   0x27       |          0
--------------+-----------------
  0x28        |          1
   0x29       |          0
   0x2a       |          0
   0x2b       |          0
   0x2c       |          0
   0x2d       |          0
   0x2e       |          0
   0x2f       |          0
--------------+-----------------
```

```
   0x30       |          0
   0x31       |          0
   0x32       |          0
   0x33       |          0
   0x34       |          0
   0x35       |          0
   0x36       |          0
   0x37       |          0
--------------+-----------------
   0x38       |          0
   0x39       |          0
   0x3a       |          0
   0x3b       |          0
   0x3c       |          0
   0x3d       |          0
   0x3e       |          0
   0x3f       |          0
--------------+-----------------
```

show qos l2-mapping-table name *table_name* : Use this command to display named table for the internal to L2 mapping values.

```
Table: l2Marktable
------------------------+-------+-------
    Internal Priority    |802.1p | MPLS
 Class-of-service | Color|       |
------------------+------+-------+-------
       0          |   0  | 0x0   |    0
       0          |   1  | 0x0   |    0
       0          |   2  | 0x0   |    0
       0          |   3  | 0x0   |    0
       1          |   0  | 0x4   |    6
       1          |   1  | 0x2   |    1
       1          |   2  | 0x2   |    1
       1          |   3  | 0x2   |    1
------------------+------+-------+-------
       2          |   0  | 0x4   |    2
       2          |   1  | 0x4   |    2
       2          |   2  | 0x4   |    2
       2          |   3  | 0x4   |    2
       3          |   0  | 0x6   |    3
       3          |   1  | 0x6   |    3
       3          |   2  | 0x6   |    3
       3          |   3  | 0x6   |    3
------------------+------+-------+-------
       4          |   0  | 0x8   |    4
       4          |   1  | 0x8   |    4
       4          |   2  | 0x8   |    4
       4          |   3  | 0x8   |    4
       5          |   0  | 0xa   |    5
       5          |   1  | 0xa   |    5
       5          |   2  | 0xa   |    5
       5          |   3  | 0xa   |    5
------------------+------+-------+-------
       6          |   0  | 0xc   |    6
       6          |   1  | 0xc   |    6
       6          |   2  | 0xc   |    6
       6          |   3  | 0xc   |    6
       7          |   0  | 0xe   |    7
       7          |   1  | 0xe   |    7
       7          |   2  | 0xe   |    7
       7          |   3  | 0xe   |    7
------------------+------+-------+-------
```



CHAPTER 41

L2 Marking Support

- Revision History, on page 313
- Feature Description, on page 313
- How it Works, on page 313
- Configuring L2 Marking Support, on page 315

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The L2 Marking Support for CUPS enables marking of QoS Class Identifier (QCI) and Differentiated Services Code Point (DSCP) derived L2 marking for CUPS. The QoS marking support is similar to the QoS marking support that is supported on the non-CUPS platform, which ensures that the QoS treatment is maintained when the packets traverse via the L2 routers.

How it Works

This section briefly describes how L2 marking works.

Basic Functionality

- The type of the L2 marking is decided at the Control Plane (CP) as per the Service-Configuration. The types of L2 marking supported are DSCP-based, QCI-based, and None.
- When the User Plane (UP) comes up with a QCI value, the lookup is performed on the associated QCI-table for the service. Based on the lookup, the priority is selected or decided for the corresponding QCI value.

- The selected Layer 2 marking type and priority is communicated to the UP in an Sx message.
- To support the passing of new information to the UP, a new custom IE is added to the FAR IE.
  - LAYER2 MARKING:
    - TYPE PRIORITY: <type> <priority-value>

      The new custom IE is defined with the type-number : 228

- When the L2 marking changes – type or priority, the same is communicated to the UP, when the bearer update occurs.

Sx Interfaces Changes

Layer 2 Marking IE in FAR

To pass the L2 Marking information to the UP for the bearer, a new custom-IE is defined and the FAR is modified to include it as follows:

Table 12: Layer 2 Marking Information Element

Information Elements	Condition / Comment	Application				IE ID
		Sxa	Sxb	Sxc	N4	
Layer2 Marking	This IE shall indicate the type of the Layer2 Marking if present.	X	X			

The Layer 2 Marking IE is encoded as follows:

Table 13: Layer 2 Marking IE Within PFCP FAR

Octet 1 and 2		Layer2 Marking IE Type = 228 (decimal)			
Octets 3 and 4		Length = n			
Information elements	Condition / Comment	Application			
		Sxa	Sxb	Sxc	N4

Octet 1 and 2		Layer2 Marking IE Type = 228 (decimal)			
Octets 3 and 4		**Length = n**			
Layer 2 Marking	This IE identifies the Layer 2 Marking to be applied for the packets matching this FAR. The length of the IE could be either 0 or 1. The 1 byte contains the following information. • TYPE PRIORITY: <type> <priority-value> • Type : (1-DSCP, 2-QCI, 3-None) - beginning 2 Bits Priority-Value: the last 6 bits • Internal-Priority in case of QCI/None type • DCSP value in case of DSCP type	X	X	Sxc	N4

Limitations

The following is the limitation for this feature in this release.

The change in the QCI table is not applied immediately to the subscriber. The change is applied only after the bearer update.

Configuring L2 Marking Support

The following section provides information about the CLI commands available to enable or disable the feature.

Configuring Internal Priority

To configure internal priority in the QCI-mapping table for the GGSN, GTPv1 P-GW, and SAEGW calls, use the following service specific configuration. This command in the GGSN service configuration overrides the behavior of QCI-QOS-mapping for data packets only.

```
configure
     context context_name
       ggsn-service service_name
          internal-qos data { dscp-derived | none | qci-derived }
          { no | default } internal-qos data { dscp-derived | none |
qci-derived }
          end
```

Notes:

- **no:** Disables the specified functionality.
- **default:** Disables the functionality.
- **dscp-derived:** Data packets are marked at Layer 2 based on DSCP configured in qci-qos mapping table, then if DSCP is not configured in the qci-qos mapping table then data packets are not marked.
- **none:** Data packets are not marked with Layer 2 (MPLS EXP/802.1P) marking.
- **qci-derived:** Data packets are marked at Layer 2 based on internal-qos-priority configured in qci-qos mapping table. If internal-qos priority is not configured in the qci-qos mapping table, then the data packets are not marked.

Associating QCI-QoS Mapping Table

Use the following commands to associate a QCI-QoS mapping table at the CP.

```
configure
  context context_name
    associate qci-qos-mapping { map_table_name map_table_name }
    exit
```

NOTES:

- **map_table_name** *map_table_name*: Specifies the name of an internal table from which to map the QoS to L2 values.

  map_table_name must be a string of 0 through 80 characters.

- This command is disabled by default.

Configuring QCI Derived L2 Marking

Use the following commands to:

- Create or modify a Layer 2 mapping table.
- Enter the QoS L2 Mapping Configuration Mode to map internal QoS priority to Layer 2 QoS values on the User Plane (UP).

```
configure
  qos l2-mapping-table { name map_table_name| system-default }
  exit
```

NOTES:

- **name** *map_table_name*: Specifies the name of QoS mapping table from which to map QoS to L2 values. It enables internal mapping to L2 values like 802.1p, mpls, and so on.

  map_table_name must be an alphanumeric string of 0 through 80 characters.

- **system-default** : Configures the system default mapping. The system default is always associated as the default for every VRF or Context.
- This command is enabled by default.

Associating L2 Mapping Table

Use the following commands to associate the configured L2 mapping table to a given VRF or Context.

```
configure
  context context_name
    associate l2-mapping-table name table_name
    exit
```

NOTES:

- **l2-mapping-table name** *table_name*: Specifies the name of an internal table from which to map QoS to L2 values.

  map_table_name must be an alphanumeric string of 0 through 80 characters.

- This command is enabled by default.

Configuring DSCP Derived L2 Marking

Use the following commands to modify the Differentiated Services Code Point (DSCP) to Class of Service (CoS) mapping on the User Plane (UP).

```
configure
  qos ip-dscp-iphb-mapping dscp dscp_value internal-priority cos
class_of_service_value
  exit
```

NOTES:

- **ip-dscp-iphb-mapping**: Manages mapping of the DSCP information in a packet to the internal QoS marking.

  “ip-dscp-iphb-mapping” is a global table per UP.

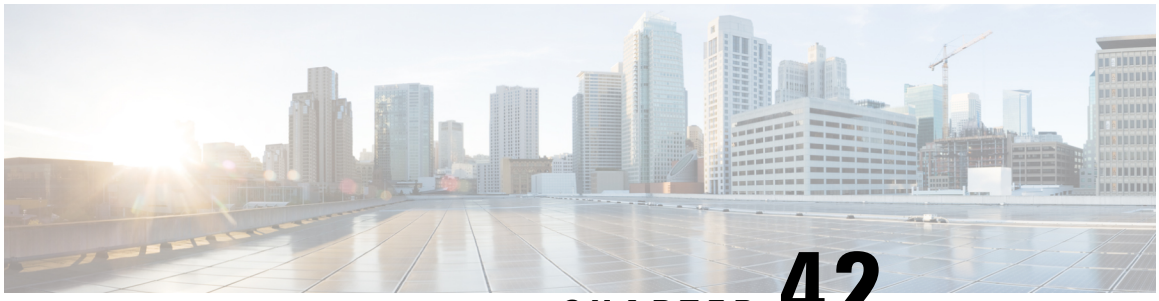
- **dscp** *dscp_value*: Maps the IP DSCP values to the internal QoS.

  dscp_value must be a hexadecimal number between 0x0 and 0x3F.

- **internal-priority cos** *class_of_service_value*: Maps to the internal QoS priority or CoS.

  class_of_service_value must be a Hexadecimal number between 0x0 and 0x7.

- This command is enabled by default.



CHAPTER 42

L3, L4, and L7 Rule Combination in Ruledef

- Revision History, on page 319
- Feature Description, on page 319
- How it Works, on page 320
- Configuring the L3, L4, and L7 Rule Combination in Ruledef Feature, on page 320
- Monitoring and Troubleshooting, on page 321

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Using the L3, L4, and L7 Rule Combination in Ruledef feature, you can allow and categorize traffic into specific Rating Group (RG) for the following:

- Specific IP addresses
- Ports
- Uniform Resource Locators (URLs)

The feature increases the scalability of the host pool from 256 to 512. The feature allows and defines a new **url-sni-pool** configuration with 256 entries in a single pool. The entries can be a mix of URL and Server Name Indication (SNI) values. The system-wide limit of URL-SNI pools is 384 entries.

How it Works

The feature enables you to define a list of URLs or SNIs for the **url-sni-pool** configuration. The system uses a pool of URLs or SNIs as an L7 filter within a ruledef. A ruledef can contain a combination of hostpool, portmap, and url-sni pool match. The system matches the url-sni-pool configuration along with the other rule lines criteria without occupying any of the 32 existing rule lines.

Configuring the L3, L4, and L7 Rule Combination in Ruledef Feature

The new URL-SNI Pool Configuration mode is available under ACS Configuration mode. Use the following configuration to enable the feature.

```
configure
   active-charging service service_name
     url-sni-pool pool_name
        http url { contains | starts-with | ends-with | = | !contains |
!starts-with | !ends-with | != } url_name
        tls sni { contains | starts-with | ends-with | = | !contains |
!starts-with | !ends-with | != } sni_identity
     ruledef ruledef_name
        ip server-ip-address host_poolname
        tcp either-port port-map port_mapname
        http-tls url-sni-pool pool_name
        end
```

Note

- The system configures the ruledef with the default all-lines AND option or **multi-line-or-all-lines** option.
- When the **url-sni-pool** rule line is configured, the URL or SNI value is always matched regardless of the AND or OR match operation.
- When the AND operation is configured, all the other rule lines is matched in addition to the URL or SNI value in the pool.
  - The AND operation is the default configuration.
- After configuring the OR operation, the system matches the following values for the rule action to take effect:
  - Any one of the other rule lines.
  - URL or SNI

Verifying the L3, L4, and L7 Rule Combination in Ruledef Feature Configuration

Use the following show CLI commands to verify the url-sni-pool configuration.

- On Control Plane: **show configuration active-charging service name** *service_name*

  For example, the following is a partial output of the show CLI command:

```
url-sni-pool url_pool1
         http url contains google.com
         tls sni contains gmail.com
```

- On User Plane: **show user-plane-service url-sni-pool name** *pool_name*

  For example, the following is a partial output of the show CLI command:

```
url-sni-pool url_pool1
         http url contains google.com
         tls sni contains gmail.com

Total url-pool(s) found: 1
```

Monitoring and Troubleshooting

Show commands and Outputs

This section provides information about the show CLI commands available in support of the feature.

show configuration active-charging service name <service_name>

Use this CLI command in Control Plane to display the url-sni-pool attachment to the ruledef.

The following is a partial sample output:

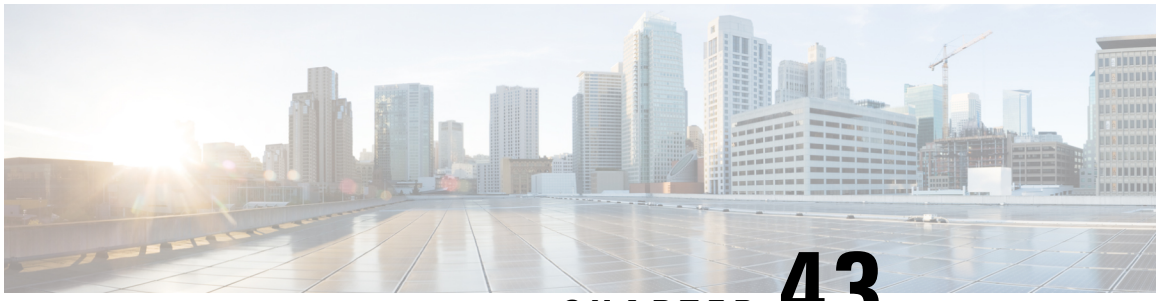
```
ruledef special_charging_group1
      ip server-ip-address range host-pool IP_FREE_MUSIC
      tcp either-port range port-map PORT_FREE_MUSIC
      http-tls url-sni-pool url_pool1
```

show user-plane-service ruledef name <ruledef_name>

Use this show CLI command in User Plane to display the url-sni-pool attachment to the ruledef.

The following is a partial sample output:

```
Ruledef Name: special_charging_group1
    ip server-ip-address range host-pool IP_FREE_MUSIC
    tcp either-port range port-map PORT_FREE_MUSIC
    Rule Application Type: Charging
    Copy Packet to Log: Disabled
    Tethered Flow Check: Disabled
    Attached Url-Sni-Pool: url_pool1
    Multi-line OR: Disabled
```



CHAPTER 43

L7 PCC Rules

- Revision History, on page 323
- Feature Description, on page 323
- How It Works, on page 324

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

With this feature, the L7 analyzer functionality is supported in the CUPS architecture.

The following L7 analyzers are supported:

- HTTP
- HTTPS
- RTP/RTSP
- FTP
- DNS
- Content Filtering
- DNS Snooping

The following charging actions are supported:

- Discard

- Terminate Flow
- Redirect (if applicable)

How It Works

This section provides a brief overview of the L7 analyzer functionality that are supported as part of this feature.

Content Filtering

Content Filtering is an in-line service available for 3GPP and 3GPP2 networks to filter HTTP requests from mobile subscribers based on the URLs in the requests. This enables operators to filter and control the content that an individual subscriber can access, so that subscribers are inadvertently not exposed to universally unacceptable content and/or content inappropriate as per the subscribers' preferences.

The Content Filtering functionality remains the same as implemented in the non-CUPS architecture. For more information, refer to *Content Filtering Support Overview* chapter in the *CF Administration Guide*.

Configuring the Content Filtering

Use the following additional configuration to enable the content filtering:

```
configure
   require user-plane content-filtering
      content-filtering category database directory path path_address
      content-filtering category database max-version version_number
      end
```

Note The above configuration must be configured on the User Plane, during boot time, to enable Content Filtering. Defining the above configuration post the User Plane configuration will lead to errors and inconsistencies.

Note To enable the feature, license for User Plane as well as existing content filtering license is required on Uplane.

Note For ICSR User Plane 1:1, the database is loaded pn both the UP's, separately. The rest of the Content Filtering configurations on Control Plane remains the same. The Content Filtering configuration is pushed from Control Plane to active the User Plane and then to standby User Plane.

Configuration on Control Plane

The following sample configuration demonstrates changes required on Control Plane for Content Filtering:

```
config
            active-charging-service ACS
               content-filtering category policy-id 1
               analyze priority 1 category ABOR
```

```
              analyze priority 2 category ADVERT action allow
             analyze priority 2 category ADVERT action allow action content-insert "Content
Restricted : The Web Guard feature has been enabled on your line. Web Guard has restricted
your access to this content. The person on your Wireless account who is designated as the
Primary Account Holder can disable this restriction through the account management website"

              exit
           rulebase cisco
              content-filtering mode category static-only
              content-filtering flow-any-error permit
              content-filtering category policy-id 5
```

The configuration on the Control Plane is pushed to User Plane using the PFD mechanism.

Use the following show commands to validate the content filtering configuration on User Plane:

- show user-plane-service rulebase name cisco
- show user-plane-service content-filtering category policy-id

Use the following show commands to check the CFDB spawning on User Plane:

- show content-filtering category database facility srdbmgr
- show content-filtering category database verbose debug-only
- show content-filtering category database verbose
- show content-filtering category database url
- show content-filtering category url

The Content Filtering policy ID received from PCRF for a particular subscriber is sent to User Plane during call establishment. The PFCP messages Sx establishment request/Sx modify request contains the CF Policy ID.

Use the following command to check the CF Policy Id on User Plane:

show subscribers user-plane-only callid full all

The following filed is displayed in support of Content Filtering in CUPS:

- Content Filtering Policy ID

Use the following show commands to monitor the SRDB Request/Response/CF Polict actions:

- show user-plane-service inline-services content-filtering category statistics
- show user-plane-service inline-services content-filtering category statistics rulebase name
- show content-filtering category statistics
- show content-filtering category statistics facility srdbmgr instance 1
- show content-filtering category statistics volume all

Note All existing bulk statistics defined for Content Filtering in the non-CUPS architecture is also applicable in CUPS.

Limitations

- Dynamic content filtering mode is not supported.
- Rulebase command **content-filtering flow-any-error [permit | deny]** is not supported.

DNS

Offloading to SM-P

DNS packets are not offloaded to SM-P.

Charging

DNS packets are charged at SM-P.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to DNS: **show user-plane-service statistics analyzer name dns**

DNS Snooping

Charging

The charging of DNS Snooping takes place at SM-P.

Rule Definitions

Use the following CLI commands for specifying the rule definition hostnames (domain-names) and part of the host names.

```
ruledef <ruledef_name>
      ip [server-domain-name {contains|=|ends-with|starts-with} <url_string>]

      ip [server-domain-name {contains|=|ends-with|starts-with} <url_string>]

       multi-line-OR enabled
```

Use the no version of this CLI to delete the ruleline for ip server- domain-name.

```
ruledef <ruledef_name>
  no ip [server-domain-name {contains|=|ends-with|starts-with} <url_string>]

   exit
```

Use the following CLI for configurable timer of DNS entries at ECS level.

```
configure
        active-charging service service_name
```

```
ip dns-resolved-entries timeout <value_secs>
end
```

Whenever the ruledef containing the ip server-domain-name keyword is defined and used in rulebase, the ip-table is created per rulebase per instance.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Show CLIs

Use the following CLIs to check the table for DNS IP entries:**show user-plane-service [statistics dns-learnt-ip-addresses {summary | sessmgr instance <id> |all [verbose] }]**

Bulkstats

The following bulkstats are available in support of DNS Snooping feature:

- ecs-dns-learnt-ipv4-entries
- ecs-dns-flushed-ipv4-entries
- ecs-dns-replaced-ipv4-entries
- ecs-dns-overflown-ipv4-entries
- ecs-dns-learnt-ipv6-entries
- ecs-dns-flushed-ipv6-entries
- ecs-dns-replaced-ipv6-entries
- ecs-dns-overflown-ipv6-entries

The above bulkstats are added in the ECS schema same as in the non-CUPS architecture.

Note The SNMP Trap generation commands are not supported in CUPS DNS snooping feature.

FTP

Offloading to SM-P

Only for FTP data, TRM is engaged. FTP data flows are eligible for offloading to SM-P.

There is no TRM engagement for control FTP flows.

Charging

FTP packets are charged at SM-P.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to FTP: **show user-plane-service statistics analyzer name ftp**

HTTP

HTTP Offloading to SM-P

On a header completion of HTTP Request/Response, the uplink/downlink data packets are offloaded to VPP in the following cases:

- Content-Length – Volume-based offloading is supported for methods like GET and POST. The HTTP flow with chunk-encoding data transfer mechanism does not get offloaded irrespective of the method defined in HTTP. If the stream is offloaded based on content-length, then the stream on the other end will also get offloaded until the former is not onloaded.
- CONNECT Method—The method where both uplink and downlink streams are offloaded after flow is upgraded to CONNECT.
- WebSocket Method—After the flow is classified as WebSocket protocol, both uplink and downlink streams are offloaded.
- The streams are onloaded back to SM-U application in either of the following cases:
  - FIN packet received.
  - Content-length is breached.
  - PDN update.

Header Parsing

Similar to non-CUPS implementation, only the header fields defined in ruledefs, which are included in rulebase, are parsed. Or, in case of features like x-header, redirection is configured which have dependencies on some of the HTTP header fields.

Rule Matching

There is no functional change in the way rule-matching takes place in CUPS. The only change is specific to TRM wherein both uplink and downlink has its own TRM.

HTTP Charging

- Complete Packets are charged at SM-P.
- Partial Packets are charged on SM-U on completion. Packet completing the Partial Packet is also charged on SM-U.
- Concatenated Packets are charged on SM-U.

- Delay Charging is enabled – In case there are uncharged bytes, the packet along with the uncharged bytes gets charged on SM-U.
- Response-based charging is enabled – On receiving a Response, both uplink and downlink packets are charged on SM-U. Subsequent uplink and downlink packets are charged at SM-P, unless they are partial/concatenated.

X-Header Parsing and Rule-Matching

Ruledefs with x-header rule-lines are parsed and matched.

WebSocket

The functionality remains the same as non-CUPS architecture.

TRM and Response-Based Charging

Transactional Rule Matching will only avoid per-packet rule matching after a flow is fully classified.

Direction-based TRM has been introduced in CUPS, wherein there are two TRMs for a flow, one for uplink and the other for downlink direction. After a packet enables TRM, subsequent packets (TRM eligible) continue to match the same rule resulting in efficient rule-matching. That is, uplink packets match the uplink TRM cached rule, and downlink packets match the downlink TRM cached rule.

URL-Based Redirection

The functionality remains the same as non-CUPS architecture.

For flow action redirect-url, encrypt is not supported. Currently, the following dynamic fields are supported:

- #HTTP.URI#
- #HTTP.HOST#
- #HTTP.URL#
- #ACSMGR_BEARER_CALLED_STATION_ID#
- #RULEBASE#
- #RTSP.URI#

X-Header Insertion

X-header Insertion is supported in HTTP Requests. The behavior remains same as that of non-CUPS architecture. With respect to offloading to SM-P:

- Flows, for which X-header is inserted in a packet, are not offloaded.
- With X-header configuration, all TCP OOO packets irrespective of transmit order CLI, will be buffered and sent out after reordering.

X-Header insertion statistics CLI

show user-plane-service statistics charging-action name *charging_action_name*

The following fields are added in support of X-header insertion:

- For Request:
  - XHeader Bytes Injected
  - XHeader Pkts Injected
  - XHeader Bytes Removed
  - XHeader Pkts Removed
  - IP Frags consumed by XHeader

Limitation

- X-Header Spoofing is not supported.
- X-Header Insertion in Response packet is not supported.
- X-Header Encryption with RSA and RC4MD5 is supported but not supported with AES.
- Monitor protocol for X-Header is not supported.
- Following X-Header fields insertion is not supported in a packet: QoS, UIDH, Customer ID, Hash Value, Time of the Day, Radius String, Session-Id, Congestion Level, User-Profile.

HTTP Analyzer Statistics

Use the following CLI command to get statistics related to the HTTP analyzer: **show user-plane-service statistics analyzer name http**

HTTPS

HTTPS Offloading to SM-P

HTTPS flows are offloaded to SM-P after receiving the application packet. With the P2P analyzer, offloading works when P2P analyzer detects the L7 protocol.

HTTPS Charging

Charging for HTTPS packets are done at SM-P.

Statistics

Use the following CLI command to get statistics related to HTTPS: **show user-plane-service statistics analyzer name secure-http**

HTTP URL Filtering

The HTTP URL Filtering feature simplifies rule definitions used for URL detection.

The HTTP request packet can have a proxy (prefixed) URL and an actual URL. If a proxy URL is found in the HTTP request packet, the HTTP URL Filtering feature truncates this URL from the parsed information and only the actual URL is used for rule matching and Event Data Records (EDR) generation.

Configuring the HTTP URL Filtering Feature

This section describes how to configure the HTTP URL Filtering feature.

Configuring Group of Prefixed URLs

To configure the group of prefixed URLs, use the following CLI commands:

```
configure
  active-charging service ecs_service_name
    group-of-prefixed-urls prefixed_urls_group_name
    end
```

Configuring URLs in the Group of Prefixed URLs

To configure URLs to be filtered in the group of prefixed URLs, use the following CLI commands:

```
configure
  active-charging service ecs_service_name
    group-of-prefixed-urls prefixed_urls_group_name
      prefixed-url url_1
      ...
      prefixed-url url_10
      end
```

Enabling the Group of Prefixed URLs in Rulebase

To enable the group of prefixed URLs in rulebase for processing prefixed URLs, use the following CLI commands:

```
configure
  active-charging service ecs_service_name
    rulebase rulebase_name
      url-preprocessing bypass group-of-prefixed-urls
prefixed_urls_group_name_1
      ...
      url-preprocessing bypass group-of-prefixed-urls
prefixed_urls_group_name_64
      end
```

This configuration on the control plane chassis will be pushed to the user plane with a PFD message for “group-of-prefixed-urls” and “rulebase-url-preprocessing”separately.

The group of prefixed URLs has the list of proxy URLs, which must be truncated. The rulebase contains multiple group of prefixed urls, which must be filtered. Charging ruledefs contain rules for actual URLs that must be searched after truncating URLs in the group of prefixed URLs.

Note

- Each group of prefixed URLs can have a maximum of ten prefixed URLs.
- A maximum of 64 group of prefixed URLs can be created and configured.

Show Commands

show user-plane-service group-of-prefixed-urls all | name *group_name*

This show command can be used on the user plane to verify whether the group of prefixed URLs are pushed or not. The output of this command is as follows:

- Name of the group of prefixed URLs
- Prefixed URLs
- Total number of prefixed URLs found

show user-plane-service rulebase name *rbase_name*

This show command can be used on the user plane to check whether the group of prefixed URLs is configured in rulebase or not. The output of this command is as follows:

- Name of rulebase
- Name of the groups of prefixed Urls for URL pre-processing

show user-plane-service statistics analyzer name http

The output of this command is as follows:

- Total HTTP Sessions
- Current HTTP Sessions
- Total Uplink Bytes
- Total Downlink Bytes
- Total Uplink Pkts
- Total Downlink Pkts
- Uplink Bytes Retrans
- Downlink Bytes Retrans
- Uplink Pkts Retrans
- Downlink Pkts Retrans
- Total Request Succeed
- Total Request Failed
- GET Requests
- POST Requests
- CONNECT Requests
- PUT requests
- HEAD requests
- Websocket Flows
- Invalid packets

- Wrong FSM packets
- Unknown request method
- Pipeline overflow requests
- Corrupt request packets
- Corrupt response packets
- Unhandled request packets
- Unhandled response packets
- Partial HTTP Header Anomaly prevented
- New requests on closed connection
- Memory allocation failures
- Packets after permanent failure
- Prefixed Urls Bypassed
- FastPath Statistics
- Total FP Flows
- Uplink (Total FP Pkts)
- Downlink (Total FP Pkts)
- Uplink (Total FP Bytes)
- Downlink (Total FP Bytes)

Note Prefixed URLs Bypassed counter has been added in http analyzer stats as a performance measurement to show the number of truncated prefixed URLs.

RTP/RTSP

Offloading to SM-P

RTP, being on UDP Protocol, is offloaded immediately.

RTSP flow is not offloaded. There is no TRM engagement for RTSP flows.

Charging

RTP packets are charged at SM-P. RTSP packets are charged at SM-P unless the packets being partial or if delay-charging is enabled.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to RTP: **show user-plane-service statistics analyzer name rtp**

Use the following CLI commands to get statistics related to RTSP:

- **show user-plane-service statistics analyzer name rtsp**
- **show user-plane-service statistics analyzer name rtsp verbose**

RTP Dynamic Flow Detection

The **rtp dynamic-flow-detection** CLI command, under the ACS Rulebase Configuration mode, enables the Real Time Streaming Protocol (RTSP) and Session Description Protocol (SDP) analyzers to detect the child RTP and RTCP flows. If you configure the RTSP/SIP and SDP analyzers, and **rtp dynamic-flow-detection** CLI is present, then there's no need for configuring RTP/RTCP explicitly. With the **rtp dynamic-flow-detection** CLI command, the child RTP or RTCP flows get corelated to their parent RTSP/SIP-SDP flows.

Once the parent flow (RTSP/SIP-SDP) gets cleared, the child RTP/RTCP flows also gets cleared. In the absence of this CLI, the L7 layer analysis for RTP and RTCP needs a separate analyzer configuration. There’s no correlation of RTP/RTCP flows to RTSP/SIP-SDP flow.

Rule-matching for Bearer-specific Filters

Rule Matching

The functionality remains the same as the non-CUPS architecture.

IMSI-based rules are matched as per the subscribers IMSI.

APN-based rules allows you to define rule expressions to match Access Point Name (APN) of the bearer flow.

RAT-Type allows you to define rule expressions to match Radio Access Technology (RAT) in the bearer flow.

Rule Definitions

Use the following CLI commands to configure the IMSI pool.

```
configure
  active-charging service service_name
    imsi-pool pool_name
      imsi { imsi_number | range start_imsi to end_imsi }
```

The imsi-pool can contain either IMSI value or range of IMSI.

Use the following CLI commands to configure rule line under ruledef.

```
configure
  active-charging service service_name
    ruledef ruledef_name
      bearer 3gpp imsi { = imsi_value } | { range imsi-pool pool_name }
      bearer 3gpp apn operator apn_name
      bearer 3gpp rat-type operator rat_type
```

IMSI range can be configured in a rule with the help of IMSI pool.

For more information about the CLI commands, see *ACS Ruledef Configuration Mode Commands* in the StarOS *Command Line Interface Reference*.

Show CLIs

Use the following CLI on User Plane to see information about IMSI pool that is configured in a service:**show user-plane-service imsipool name** *pool_name*

SIP

Offloading to SM-P

SIP flow is not offloaded.

Charging

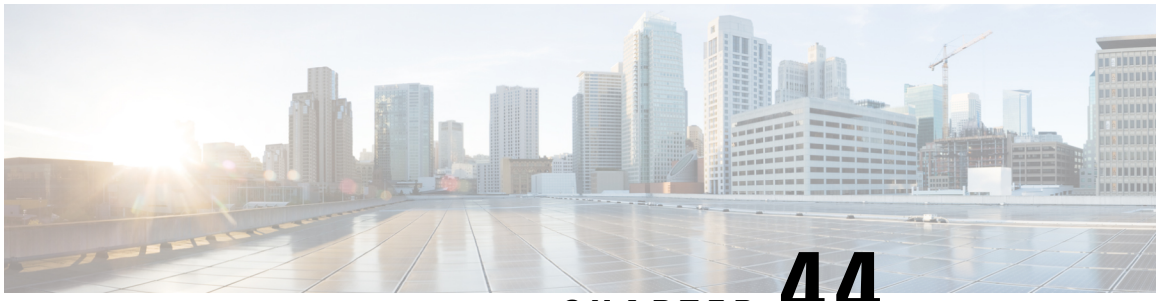
SIP packets are charged at SM-P.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to SIP: **show user-plane-service statistics analyzer name sip**



CHAPTER 44

Local Policy in CUPS

- Revision History, on page 337
- Feature Description, on page 337
- How It Works, on page 338
- Configuring Local Policy in CUPS, on page 338

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The local policies are used to control different aspects of a sessions such as - QoS, Data Usage, Subscription profiles, Server Usage, and so on, by means of locally defined policies. It is intended as a replacement or enhancement to PCRF-based policy control. The local policies are triggered during certain events and the associated conditions.

The Local Policy functionality has the following advantages:

- Reusability: Reusable rules engine as a common infrastructure for PCRF-based policies.
- Resource Consumption: Lower memory usage, CPU usage and response time.
- Extensibility: Extensible to handle new events and attributes with minimal effort.
- Execution speed: Shorter reaction time for network events.
- Integration: Seamless integration with the existing policy infrastructure - IMSA and PCEF with a minimal impact on existing services. In case of unreachable events, a mechanism to fallback to PCRF is implemented.

Local policies are useful in various scenarios. For example:

- A Local Policy operates as a fallback mechanism when PCRF is unavailable or when an operator has not deployed PCRF in the infrastructure.
- As an enhancer to PCRF triggers, handling certain triggers locally or to handle triggers unsupported by 3GPP Standards or PCRF.
- Deployments where the subscription policies are static and tiered or has well defined subscriber groups.
- When the response time required is less.

Note The working of the Local Policy feature in the CUPS environment is similar to the non-CUPS P-GW and SAEGW nodes.

How It Works

Local Policy feature is implemented based on the following concepts:

- Event driven rules engine. For example, RAT change event.
- On a registered Event Trigger occurrence, series of registered rules are evaluated based on the Type of Event and the current State.
- On a successful rule match, series of actions are executed.

Configuring Local Policy in CUPS

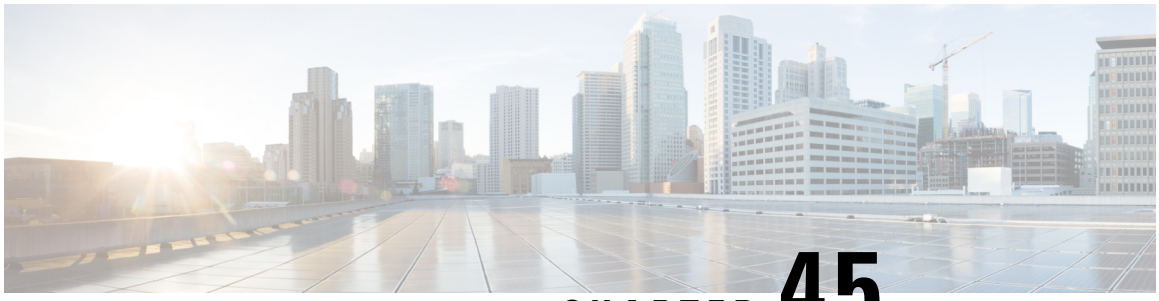
Note The CLI commands available for non-CUPS Local Policy feature are also applicable in CUPS environment.

Following is a sample Local Policy configuration in Control Plane node:

```
configure
   local-policy-service service_name
      ruledef ruledef_name
         condition priority priority radio-access-technology eq eutran
      ruledef ruledef_name
         condition priority priority apn eqcompare_string
      actiondef actiondef_name
         action priority priority default-qos qci qci_value arp arp_value
      actiondef actiondef_name
        action priority priority activate-rulebase name rulebase_name eventbase
 eventbase_name
         rule priority priority event new-call ruledef ruledef_name actiondef
actiondef_name
         rule priority priority event location-change ruledef ruledef_name
```

```
actiondef actiondef_name
        end
```

Note No configuration is required in User Plane node.



CHAPTER 45

Load/Overload and UP Data Throttling Support on Sx

- Feature Description, on page 341
- How It Works, on page 341
- Configuring Load and Overload Support, on page 343
- Monitoring and Troubleshooting, on page 347

Feature Description

The Load/Overload support is implemented in the UPC CUPS architecture. This support is handled between the Contol Plane (CP) and User Plane (UP).

Load control enables UP to send its load information to CP to adaptively balance the PFCP session load across the UP functions according to their effective load, whereas Overload control enables throttling of new session requests towards a particular UP.

How It Works

User Plane Selection

When Load/Overload support is enabled, UP selection is implemented as given below, with a UP group:

- If none of the UP is in overload condition, Load Control Information (LCI) is used for UP selection. In this case, the least loaded UP will be selected.
- If all UPs are in the overload state, UP selection is based on the Overload Control Information (OCI). In this case the least overloaded UP is selected.
  - After a particular UP is selected, the reduction metric is still applied to this UP for throttling.
  - If throttling needs to be dropped, UP selection request is rejected for that PDN connection.
- In some scenarios where some of the UPs are in Overload condition and some of the UPs are not in Overload condition, the selection is done based on the OCI value.

- If the LCI or the OCI value are the same for a peer node, the session count information is used for UP selection.

Node-level Load/Overload Support

The CP informs the UP about the Load/Overload support enabled. Based on this information UP decides to send the Load/Overload information towards CP peer or not.

Load/Overload support at CP is configured as part of the Sx-Service node configuration. This information is sent to the UP during Sx Association Response or Sx Association Update request, if the information has changed through dynamix configuration.

Note If CP does not support Load/Overload feature through supported CLI, it ignores the reported Load/Overloadby UP. In that case, UP selection continues with the session count information.

Sx Establishment Request Throttling at CP in Overload State

Once the UP is in Overload situation, CP starts throttling the Sx Establishment Request message towards UP. This avoids new calls (Low priority/non-emergency) towards the overloaded UP.

Throttling happens based on reported OCI values – Overload Reduction metric value. The value is calculated in percentage. It randomly drops the required percentage of Sx Establishment Request towards that UP Peer. This results in call drop at CP with "sx-no-resource" disconnect reason. Also, respective statistics are incremented for the same.

Note The eMPS (high priority) subscribers' Session/ Emergency Subscribers' session is not throttled.

Sx Establishment Request Throttling at UP in Self-Protection

Once the UP is in Self-Protection state, it starts rejecting all the new sessions (non-eMPS session only), Sx Establishment Request, and Sx Modification Request for the existing sessions (non-eMPS session only).

Session Termination Trigger from UP in Self-Protection

Being in Self-Protection mode, if there are no improvement in the Load condition at UP, it starts triggering Session Termination Request towards CP in a staggered manner through Sx Report Request message indicating that UP is in Self-Protection. Based on this, CP starts initiating Sx Termination Request for those sessions.

Self Protection Termination Request (SPTER): This bit is set from UP towards CP for initiating Self-Protection based termination. The CP releases the call with disconnect reason as "graceful-term-up-self-protectn".

Note When Actual Load value is greater than the Session-Termination-Start-Threshold value, Session termination is triggered towards CP.

Limitation

This feature has the following known limitations:

- The maximum number of Load/Overload profiles supported on UP is 8.
- If the Load/Overload profiles are not configured in all UPs in the UP group, it can lead to uneven distribution of sessions. It is recommended that all UPs must be configured with the Load/Overload support in a single UP group.
- After session recovery, SessMgr instance gets to relearn Load/Overload values from SxDemux. The SxDemux communicates these values only when there is a change in Load/Overload values.
- Toggling the Load Control configurations on the fly (enable to disable or disable to enable) is not supported.
- All UPs in a UP group should either be enabled or disabled. Having one of the UP enabled and another as disabled in a UP group is not supported as it can result in improper Load/Overload values on CP.
- It's recommended to disable Load Control for all IMS UPs.

Configuring Load and Overload Support

The Load and Overload support is configured using the following configurations:

- User Plane Load Control Profile Configuration
- User Plane Overload Control Profile Configuration
- Association of Load Profile to a User Plane Service
- Sx Protocol Configuration on Control Plane

User Plane Load Control Profile Configuration

Use the following commands to configure load control profile.

```
configure
  userplane-load-control-profile profile_name
  end
```

Configuring User Plane Load Control Profile Parameters

Use the following configuration to configure UP Load profile parameters:

```
configure
  userplane-load-control-profile profile_name
     system-weightage system-cpu-utilization utilization_value
system-memory-utilization utilization_value license-session-utilization
utilization_value
     sessmgr-weightage sessmgr-cpu-utilization utilization_value
system-memory-utilization utilization_value
    inclusion-frequency advertisement-interval interval_value change-factor
```

```
changefactor_value
    end
```

NOTES:

- **inclusion-frequency**: Configures parameters to decide inclusion frequency of load control information IE.
- **advertisement-interval**: Advertisement interval is the periodic interval after which the load value is advertised. Configures the advertisement interval for load control. The default value is 300. Set the value as 0 (zero) to include LCI IE in every applicable message.
- **change-factor**: Change-factor is the delta increase or decrease in the load value based on which the load advertisement occurs. Configures the change factor for load control. The default value is 5.
- **sessmgr-weightage**: Configures sessmgr weightage for various load control parameters. Total weightage of all the parameters should be 100. The default ratio is 65% weightage to sessmgr-cpu-utilization and 35% weightage to sessmgr-memory-utilization.
- **sessmgr-cpu-utilization**: Configures session manager CPU utilization weightage in percentage. Default weightage in load factor is 35%.
- **sessmgr-memory-utilization**: Configures session manager memory utilization weightage in percentage. Default weightage in load factor is 65%.
- **system-weightage**: Configures system weightage for various load control parameters. Total weightage of all the parameters should be 100. The default values are 40% weightage to system-cpu-utilization, 30% weightage to system-memory-utilization and 30% weightage to license-session-utilization.
- **system-cpu-utilization**: Configures system CPU utilization weightage in percentage. Default weightage in load factor is 40%.

Note The value displayed in the **show cpu table** CLI command is based on the average value of 5 minutes, 10 minutes, and 15 minutes. Use the result of the average value for system CPU utilization to verify the utilization manually.

- **system-memory-utilization**: Configures system memory utilization weightage in percentage. Default weightage in load factor is 30%.

Note The value displayed in the **show cpu table** CLI command is based on the average value of 5 minutes, 10 minutes, and 15 minutes. Use the result of the average value for system memory utilization to verify the utilization manually.

- **license-session-utilization**: Configures license session utilization weightage for User Plane service in percentage. Default weightage in load factor is 30%. The license utilization percentage is equal to the utilization percentage of the current UP sessions out of the maximum UP sessions.

User Plane Overload Control Profile Configuration

Use the following commands to configure overload control profile.

```
configure
  userplane-overload-control-profile profile_name
  end
```

Configuring User Plane overload Control Profile Parameters

Use the following configuration to configure UP overload profile parameters:

```
configure
  userplane-overload-control-profile profile_name
    overload-threshold system lower-limit limit_value upper-limit limit_value
 sessmgr lower-limit limit_value upper-limit limit_value vpp-cpu lower-limit
 limit_value upper-limit limit_value
     system-weightage system-cpu-utilization utilization_value
system-memory-utilization utilization_value license-session-utilization
utilization_value
     sessmgr-weightage sessmgr-cpu-utilization utilization_value
system-memory-utilization utilization_value
    inclusion-frequency advertisement-interval interval_value change-factor
 changefactor_value
     tolerance tolerance_value
     validity-period validity_period
     end
```

NOTES:

- **inclusion-frequency**: Configures parameters to decide inclusion frequency of overload control information IE.
- **advertisement-interval**: Advertisement interval is the periodic interval after which the overload value is advertised. Configures the advertisement interval for overload control. The default value is 300. Set the value as 0 (zero) to include LCI IE in every applicable message.
- **change-factor**: Change-factor is the delta increase or decrease in the overload value based on which the overload advertisement occurs. Configures the change factor for overload control. The default value is 5.
- **tolerance**: Configures the Overload tolerance limits.
- **validity-period**: Configures validity of overload control information. Default value is 600.
- **overload-threshold**: Configures Overload thresholds limits for system, sessmgr and vpp-cpu.
- **system**: Configures overload system threshold after which node moves to self-protection mode.
- **vpp-cpu**: Configures the overload vpp-cpu threshold after which node moves to self-protection mode.
- **sessmgr**: Configures the overload threshold for session manager after which node moves to self-protection mode.
- **upper-limit** *limit_value*: Configures the various upper limit values. Following are the various upper limit values:
  - System Threshold Upper Limit : Configures overload system threshold after which node moves to self-protection mode. Default limit value is 80%.

- Sessmgr Threshold Upper Limit : Configures overload SessMgr threshold after which node moves to self-protection mode. Default limit value is 60%.
- vpp-cpu Threshold Upper Limit : Configures overload vpp-cpu threshold L2 after which node moves to self-protection mode. Default limit value is 60%.

- **lower-limit** *limit_value*: Configures the various lower limit values. Following are the various lower limit values:
  - System Threshold Lower Limit: Configures overload system threshold after which node moves to self-protection mode. Default limit value is 60%.
  - Sessmgr Threshold Lower Limit: Configures overload SessMgr threshold after which node moves to self-protection mode. Default limit value is 50%.
  - vpp-cpu Threshold Lower Limit : Configures overload vpp-cpu threshold L1 after which node moves to self-protection mode. Default limit value is 50%.

- **sessmgr-weightage**: Configures sessmgr weightage for various overload control parameters. Total weightage of all the parameters should be 100. The default ratio is 65% weightage to sessmgr-cpu-utilization and 35% weightage to sessmgr-memory-utilization.
- **sessmgr-cpu-utilization**: Configures session manager CPU utilization weightage in percentage. Default weightage in overload factor is 35%.
- **sessmgr-memory-utilization**: Configures session manager memory utilization weightage in percentage. Default weightage in overload factor is 65%.
- **system-weightage**: Configures system weightage for various overload control parameters. Total weightage of all the parameters should be 100. The default values are 40% weightage to system-cpu-utilization, 30% weightage to system-memory-utilization and 30% weightage to license-session-utilization.
- **system-cpu-utilization**: Configures system CPU utilization weightage in percentage. Default weightage in overload factor is 40%.

Note The value displayed in the **show cpu table** CLI command is based on the average value of 5 minutes, 10 minutes, and 15 minutes. Use the result of the average value for system CPU utilization to verify the utilization manually.

- **system-memory-utilization**: Configures system memory utilization weightage in percentage. Default weightage in overload factor is 30%.

Note The value displayed in the **show cpu table** CLI command is based on the average value of 5 minutes, 10 minutes, and 15 minutes. Use the result of the average value for system memory utilization to verify the utilization manually.

- **license-session-utilization**: Configures license session utilization weightage for User Plane service in percentage. Default weightage in overload factor is 30%. The license utilization percentage is equal to the utilization percentage of the current UP sessions out of the maximum UP sessions.

Associating a Load Control Profile with a User Plane Service

Use the following commands to associate the Overload Control profile to a use plane service.

```
configure
   context context_name
      user-plane-service service_name
      [ no ] associate userplane-load-control-profile profile_name
```

NOTES:

- **associate**: This command associates the user plane overload control profile with a user plane service.

Sx Protocol Configuration on Control Plane

The CP Function Features IE indicates the features supported by CP. Only features having an impact on the (system-wide) UP function behaviour are signalled in this IE.

The following features are supported by CP:

- LOAD (Load Control)
- OVRL (Overload Control)

Use the following configuration to configure the supported features on CP through the Sx Protocol:

```
configure
   context context_name
      sx-service service_name
        sx-protocol supported-features { load-control | overload-control
}
       no sx-protocol supported-features [ load-control | overload-control
 ]
        end
```

NOTES:

- **supported-features**: Configures supported features for Sx interface by CP. Default value is Disabled.
- **load-control**: Enables or disables Load control feature support on CP function.
- **overload-control**: Enables or disables the Overload control feature on CP function.

Monitoring and Troubleshooting

Show Commands Input and/or Outputs

This section provides information regarding show commands and their outputs in support of the feature.

show userplane-load-control-profile name *name*

The following fields are displayed in support of this feature:

- User Plane Load Control Profiles
- User Plane Load Control Profile Name
- System Weightage and Thresholds:
  - CPU Utilization Weightage
  - Memory Utilization Weightage
  - License Session Utilization Weightage
  - System Threshold Lower Limit
  - System Threshold Upper Limit
- Sessmgr Weightage and Thresholds:
  - CPU Utilization Weightage
  - Memory Utilization Weightage
  - Sessmgr Threshold Lower Limit
  - Sessmgr Threshold Upper Limit
- VPP Weightage and Thresholds:
  - VPP Utilization Weightage
  - vpp-cpu Threshold Lower Limit
  - vpp-cpu Threshold Upper Limit
- Inclusion Frequency:
  - Change Factor
  - Advertisement Interval

show userplane-overload-control-profile name *name*

The following fields are displayed in support of this feature:

- User Plane Overload Control Profiles
- User Plane Overload Control Profile Name
- System Weightage and Thresholds:
  - CPU Utilization Weightage
  - Memory Utilization Weightage
  - License Session Utilization Weightage
  - System Threshold Lower Limit
  - System Threshold Upper Limit

- Sessmgr Weightage and Thresholds:
  - CPU Utilization Weightage
  - Memory Utilization Weightage
  - Sessmgr Threshold Lower Limit
  - Sessmgr Threshold Upper Limit
- VPP Weightage and Thresholds:
  - VPP Utilization Weightage
  - vpp-cpu Threshold Lower Limit
  - vpp-cpu Threshold Upper Limit
- Inclusion Frequency
  - Change Factor
  - Advertisement Interval
- Validity Period

show user-plane-service statistics all

The following fields are displayed in support of this feature:

- Overload Stats
  - Current State : Normal
  - Number of time self-protection condition reached in user plane : 0
  - No of Session Eshtablishment Req rejected during self-protection mode : 0
  - No of Session Modif Req rejected during self-protection mode : 0
  - No of eMPS Session Eshtablishment Req allowed during self-protection mode : 0
  - No of eMPS Session Modif Req allowed during self-protection mode : 0
  - Overload reduction metric : 0
  - Current Overload factor system : 0
  - Current Overload factor sessmgr : 0
  - Current Overload factor vpp cpu : 0
- Overload Data Stats:
  - Total Packets dropped due to overload : 0
  - Total Bytes dropped due to overload : 0
  - Total Packets dropped in self-protection mode : 0

- Total Bytes dropped in self-protection mode : 0

- Load Stats:
  - Load metric : 0
  - Current Load factor system : 0
  - Current Load factor sessmgr : 0
  - Current Load factor vpp cpu : 0

- eMPS PDNs Total
  - Active
  - Setup
  - Released
  - Rejected

show sx service statistics all

The following fields are displayed in support of this feature:

- Throttled

Bulk Statistics

Following bulkstats are available in support of Load and Overload Support on Sx feature.

Table 14: Supported Bulk Stats

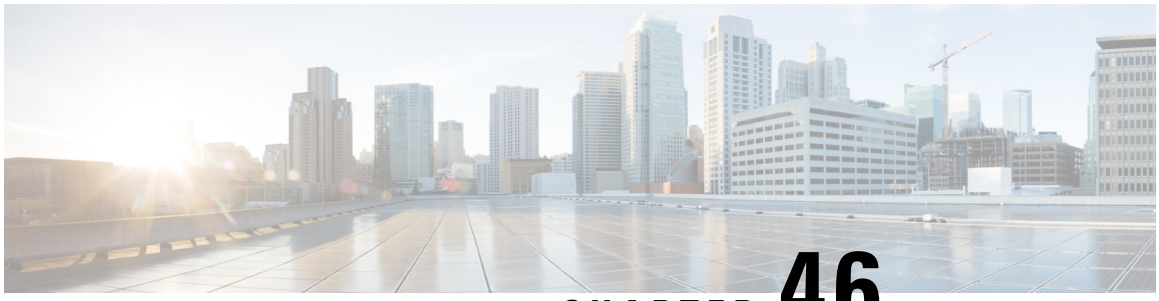
Bulkstats	Description
num-self-protection-reached	Total number of time self-protection condition reached in UP.
num-session-estab-rejected-on-self-protection	Total number of Session Establishment Request rejected during self-protection mode.
num-session-modif-rejected-on-self-protection	Total number of Session Modification Request rejected during self-protection mode.
num-emps-session-estab-allowed-on-self-protection	Total number of eMPS Session Establishment Request allowed during self-protection mode.
num-emps-session-modif-allowed-on-self-protection	Total number of eMPS Session Modification Request allowed during self-protection mode.
overload-reduction-metric	Overload reduction metric is calculated based on the configured Lower and Upper limit of Overload Condition.

Bulkstats	Description
overload-factor-system	Overload factor system is calculated based on the System CPU, Memory, VPP CPU, and other information polled from Resource Manager (RM).
overload-factor-session	UP starts rejecting new sessions and data throttling during self-protection mode.
overload-factor-vpp-cpu	Total average VPP CPU per core during overload.
load-metric	Total number of current Load metric.
load-factor-system	Total number of current sytem load factor.
load-factor-session	Total number of current session load factor.
load-factor-vpp-cpu	Total number of current VPP CPU load factor.
num-packets-dropped-on-overload	Total number of packets dropped during the overload.
num-bytes-dropped-on-overload	Total number of bytes dropped during the self protection mode.
num-packets-dropped-on-self-protection	Total number of packets dropped during the self protection mode.
num-bytes-dropped-on-self-protection	Total number of bytes dropped during the self protection mode.

SNMP Traps

The following SNMP Traps are added in support of this feature:

- UPlaneSelfOverload: When system enters into Self-Protection mode.
- UPlaneSelfOverloadClear: When system is out of Self-Protection mode.



CHAPTER 46

LTE - Wi-Fi Seamless Handover in CUPS

- Revision History, on page 353
- Feature Description, on page 353
- How It Works, on page 354
- Configuring LTE and Wi-Fi Seamless Handover, on page 355
- Monitoring and Troubleshooting, on page 356

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Seamless handovers between LTE and Wi-Fi (S2a/S2b), for UEs that need continuity with their ongoing data session, is supported in the CUPS architecture.

When handover is initiated from LTE to Wi-Fi, the Delete Bearer Request (DBR) is sent over the LTE tunnel immediately when the Create Session Response (CSR) is sent on the Wi-Fi tunnel. This causes some packet loss because of the IPSec tunnel establishment delay at the ePDG. To address the issue of packet loss, a Delete Bearer Request is sent on LTE tunnel only on expiry of the configured handover timer. If the LTE tunnel is active, uplink and downlink data are exchanged on the LTE tunnel. When handover is complete, uplink and downlink data is exchanged on the Wi-Fi tunnel. This prevents packet loss. During Wi-Fi to LTE handover, if the Modify Bearer Request is received with HI=1, it initiates a tunnel switch from Wi-Fi to LTE as per the specification.

With this feature, the following benefits are seen:

- Minimum packet loss during LTE to Wi-Fi (S2bGTP) handover and making the handover seamless (that is, MAKE before BREAK).

- LTE procedures are handled gracefully over the LTE tunnel when both tunnels are established with the P-GW.
- Wi-Fi procedures are handled gracefully over the Wi-Fi tunnel when both tunnels are established with the P-GW.

Important

- In an LTE to Wi-Fi or Wi-Fi to LTE handover, a tunnel identifier is allocated for new access traffic type for experiencing seamless handover.

How It Works

LTE - Wi-Fi Handover

- Before HO is started:
  - In case of multiple outstanding CCR-Us being supported, all requests before the hand-off requests are dropped.
  - Any pending transactions on LTE access are discarded. For example, if CBR or UBR is sent for LTE access and hand-off is initiated before completion of CBR or UBR transaction, then CBR or UBR is ignored at the P-GW. PCRF is not notified about failure.
- During the transition period:
  - If PCRF sends RAR for policy change, it is processed after handover is complete.
  - If ASR is received, then call drop occurs and both tunnels go down.
  - If session-release occurs from PCRF, then call is dropped and CSR is sent with cause as “no-resources”.
  - If the user moves back to LTE (that is, recurring handoff from LTE to Wi-Fi to LTE) with HO-Ind set to 1 (after guard timer), then the HO is processed successfully and user session is moved to LTE again.
  - If the user moves back to LTE (that is, recurring handoff from LTE to Wi-Fi to LTE) with HO-Ind set to 0, then it leads to context replacement. Old call is cleared on Wi-Fi access with the reason "Context Replacement", and the call is processed like a new call over LTE.
  - If Modify Bearer Command (MBC) is received in LTE (New access), it is rejected with Service-Denied message.
  - If Modify Bearer Command (MBC) is received in Wi-Fi (Old access), it is discarded.
  - If Delete Bearer Command (DBC) is received in LTE (New access) during the HO in progress, session is terminated.
  - In case of Sx Path Failure during an ongoing handover, on-going transactions are aborted, resulting in tearing down the call locally.
  - GTPC S5/S11 path failure

- During LTE to Wi-Fi HO, if path failure occurs on an older tunnel, then the call is cleared. If path failure occurs on a newer tunnel, it result in tearing the call .
- During the Wi-Fi to LTE HO, when path failure happens on an older tunnel, the older tunnel is cleared and the new tunnel call continues. This is possible only if the MBReq is pending from MME. In all other states, the call is teared down locally.
- WIFI to LTE (Collapsed call) HO, call continuation is not possible. Path failure on an older tunnel only results in tearing down the call locally.
- During the HO, if path failure occurs on a Newer tunnel, it will result in tearing down the call.

ICSR and Session Recovery

- At Control Plane, during transition, the most recent is considered as the stable state and a full checkpoint is triggered once handover is complete from LTE to Wi-Fi (S2BGTP) or vice-versa. This is applicable to Session Recovery and ICSR. User Plane has individual session recovery and ICSR check pointing on every message received.
- During handover failure, that is, when CP and UP are out of sync, the CP session is recovered on the most recently accessed state and UP is recovered in the new transition state. This behavior is applicable during UP failure.

Limitations

The LTE - Wi-Fi Seamless Handover feature does not support LTE to eHRPD and Wi-Fi to eHRPD handover and hand back.

Standards Compliance

The LTE – Wi-Fi Seamless Handover feature is compliant with the following standards:

- 3GPP TS 23.214
- 3GPP TS 29.244
- 3GPP TS 23.401
- 3GPP TS 23.402

Configuring LTE and Wi-Fi Seamless Handover

The following section provides information about the CLI commands available to enable or disable the feature.

Use the following CLI commands to configure LTE to Wi-Fi handover timer.

```
configure
  context context_name
    apn apn_name
      lte-s2bgtp-first-uplink timeout_value
```

```
{ default | no } lte-s2bgtp-first-uplink
end
```

NOTES:

- **default**: Enables the LTE to Wi-Fi handover completion to occur when the Create Session Response is sent on the Wi-Fi tunnel.
- **no**: Disables the feature and handover completion occurs on Create Session Response.
- **lte-s2bgtp-first-uplink** *timeout_value*: Configures LTE to S2bGTP handover completion timeout in multiples of 100 milliseconds. The valid range is from 100 to 3000. The recommended configuration is 1000 milliseconds.
- By default, the LTE to Wi-Fi handover completion happens when Create Session Response is sent on the Wi-Fi tunnel. However, after handover timeout is configured, the handover is delayed until timeout.
- Triggering handover based on first uplink data packet is not supported because the User Plane and Control Plane nodes are separated in the CUPS architecture.

Monitoring and Troubleshooting

This section provides information regarding CLI commands available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show apn statistics name <*name*>

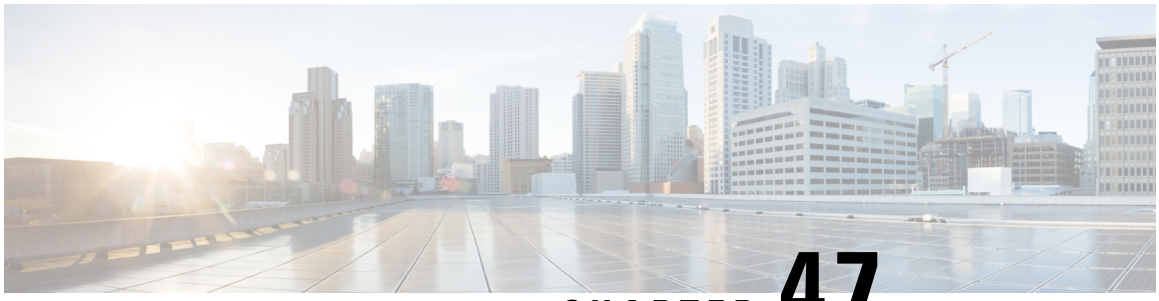
The output of this CLI command has been enhanced to display the following new fields for the APN:

- LTE-to-S2bGTP handover Succeeded on Timer Expiry – Specifies the number of handovers due to timer expiry.

NOTES:

The new fields, introduced as part of this feature, are also displayed for the following CLI commands:

- **show pgw-service statistics name** *service_name* **verbose**
- **show pgw-service statistics name all verbose**
- **show saegw-service statistics all function pgw verbose**



CHAPTER 47

Monitor Subscriber for CUPS

- Revision History, on page 357
- Feature Description, on page 357
- Monitor Subscriber Sx Private IE, on page 359
- Control Plane SMGR Functionality, on page 363
- User Plane SMGR Functionality, on page 363
- Multi PDN Multi Trace, on page 364
- MonSub Stats, on page 365
- X-Header, on page 365
- How It Works, on page 365
- Configuring the Hexdump Module for MonSub in UPF, on page 372
- Monitoring and Troubleshooting, on page 374

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The Monitor Subscriber (MonSub) feature enables tracing of subscriber-related information which includes user and control traffic, and events such as charging and internal events that are useful for debugging. By default, this information is displayed on the Control Plane console, where you can execute MonSub tracing CLI command, and captured in a Packet Capture (PCAP) file on the User Plane.

User traffic is carried on slowpath where packets traverse to the application or fastpath where packets do not have to traverse up to the application but are offloaded to fastpath processing (VPP). Slowpath mode was the default mode until fastpath offload (VPP) into SAEGW, was introduced.

Monitor Subscriber provides the following functionality:

- Continuous capture of user traffic from fastpath in PCAP files on the User Plane.
- The non-user traffic information, that is, control event traffic and other related information are displayed in Control Plane console and are captured in separate PCAP files on the User Plane.
- New option UP PCAP trace [W - UP PCAP Trace (ON)] is introduced for CUPS on Control Plane and User Plane in MonSub CLI. The new option is like the D option in the ICUPS. The slow-path and fast-path PCAP generates only when this option is ON.
- There are a maximum of four subscriber tracing sessions per NPUMGR instance. The NPUMGR (per User Plane instance) enforces the maximum tracing session limit. Slow-path capture naming convention contains the MonSub tracing session ID on SMGR instance, whereas fast-path tracing session contains the PSN as session ID. If there are already four tracing sessions running at SESSMGR instance, then slow-path capture is by name “S4”. It continues until the time NPUMGR rejects the tracing session due to max tracing limit reached.

Following are some of the important definitions that are related to this feature:

- **Chassis Traffic Volume**: The total volume of packet throughput on the chassis.
- **Monitored Traffic Volume**: Monitoring of the total throughput of all the subscribers through MonSub across all the MonSub sessions.
- PCAP success: The percentage of the MonSub traffic capture request and the successful capture in the PCAP files.

Packet Processing Throughput

Following are the scenarios that impact the packet processing throughput:

- When VPP utilization is above 80%, MonSub may have an impact to packet processing throughput. The impact is in proportion to the monitored traffic volume.
- Specifically, when the monitored traffic volume approaches 10% of the chassis traffic volume, there may be an impact on the VPP throughput causing subscriber packet loss.
- The impact to packet processing throughput is higher when using monitor priorities above 0 (zero).

Caution You must be cautious during the packet processing. When VPP is running at 80% utilization and handling approximately 10-Gbps chassis traffic volume, there’s an impact on the packet processing, if the set of MonSub sessions is collectively monitoring the subscribers, totaling more than 1 Gbps of monitored traffic volume.

PCAP Success

The PCAP success depends on the following factors:

- The level of PCAP success depends on several factors, including monitored traffic volume, VPP utilization, MonSub monitor priority, and background disk I/O.
- In general, the PCAP success rates are greater for the following cases:
  - When the VPP utilization is low and/or MonSub monitor priority is above best-effort.

- When the monitored traffic volume is less than 10% of the chassis traffic volume.

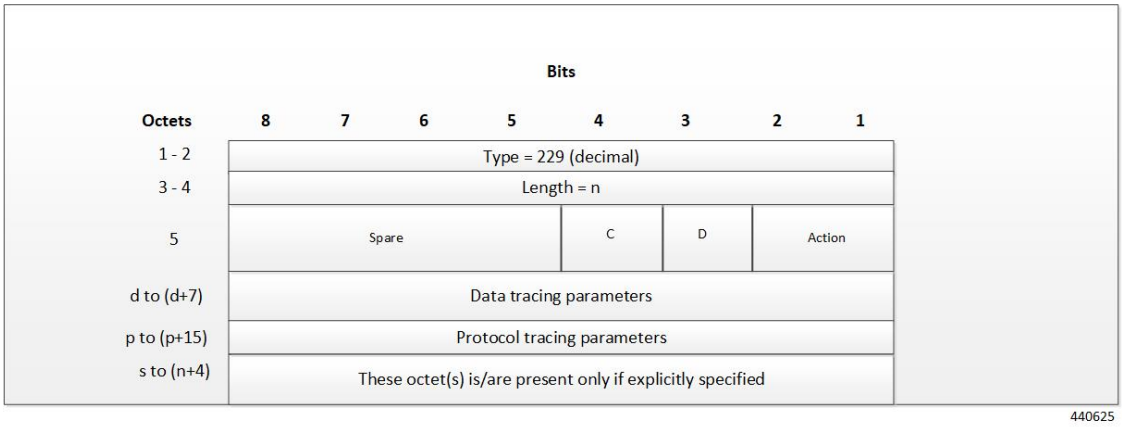
Example: When VPP is running at 80% utilization, handling approximately 10-Gbps chassis traffic volume, monitored traffic volume up to 1 Gbps is likely to yield high PCAP success percentages.

Monitor Subscriber Sx Private IE

SUBSCRIBER TRACE

The Monitor Subscriber Sx Private IE is conditional IE in the Sx Session Establishment Request and Sx Session Modification Request. This IE is valid for Sxa, Sxb and Saxb call types only.

Figure 12: Subscriber Tracing



Action: STOP / START monitor subscriber tracing. STOP =1, START =2.

Note D = DATA events tracing is ON if D=1. The 8 octets (d to d+7) contain data events tracing information should be present only when D=1.

C = CONTROL events tracing is ON if C=1.

Data Tracing Information (8 octets): It will contain the data filter parameters like Packet capture, Packet capture size, and MEH header.

- Octet 1:
  - Bit 1 – VPP enable/disable
  - Bit 2 – FCAP - Packet capture
  - Bit 3 – MEH present
  - Bit 4 to 6 - Priority
- Octet 2 to 3: Packet size

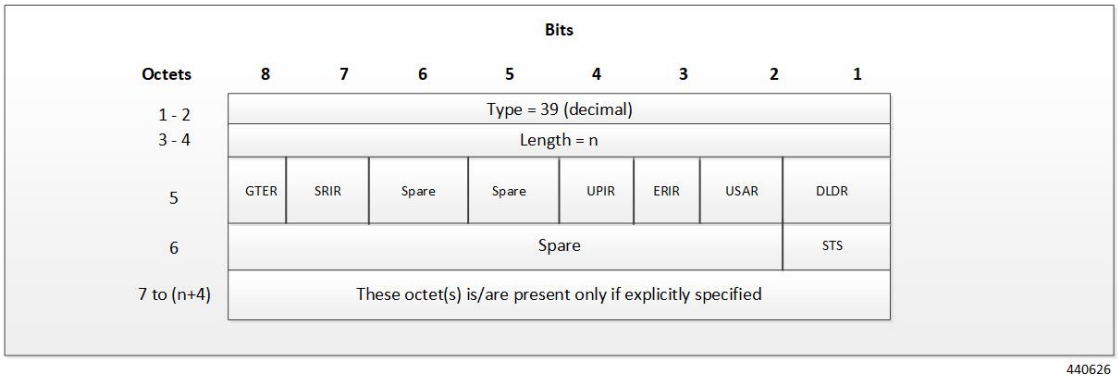
- Octet 4 – 8: Reserved for future use. Currently, all set to 0.

Protocol Tracing Information (16 octets/128 bits): The 16 octets (p to p+15) contain protocol tracing information and should be present only when either control flag (C) or data flag (D) is enable. Each bit represents a unique protocol to monitor. Example, If 49th bit is 1, PFCP events tracing is ON. The Protocol Tracing *Rulematch Events (Option 34)*, *L3 Data (Option 19)*, *EDR (Option 77)* and *Subscriber Summary After Call Disconnect* are controlled by control event flag.

Subscriber Trace Status Report (UP to CP only)

When Subscriber Trace is enabled for a PFCP session, the Report Type IE contains one extra octet (Octet 6). Presence of this octet is indicated by the length.

Figure 13: Report Type IE



Octet 5 shall be encoded as follows:

- Bit 1 – DLDR (Downlink Data Report): when set to 1, this indicates Downlink Data Report.
- Bit 2 – USAR (Usage Report): when set to 1, this indicates a Usage Report .
- Bit 3 – ERIR (Error Indication Report): when set to 1, this indicates an Error Indication Report.
- Bit 4 – UPIR (User Plane Inactivity Report): when set to 1, this indicates a User Plane Inactivity Report.
- Bit 5–6 Spare.
- Bit 7 – SRIR (Session Replacement): when set to 1, this indicates a Session Replacement request from UP.
- Bit 8 – GTER (Graceful termination): when set to 1, this indicates a Graceful Termination request from UP.

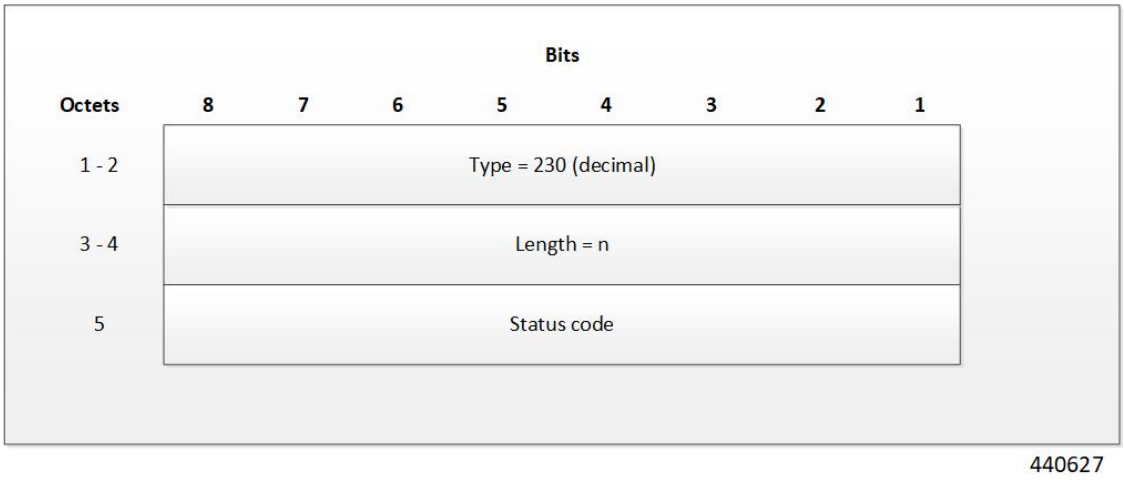
Octet 6 (present when Length>1) to be encoded as follows:

- Bit 1 – STS (Subscriber Trace Status Report): when set to 1, this indicates Subscriber Trace Status Report.
- Bit 2 to 8 – Spare.

Subscriber Trace Status Report IE (Private IE)

The Subscriber Trace Status Report IE is a conditional IE for only Sxa, Sxb and Sxab call types. For N4 call type, this IE is not present.

Figure 14: Subscriber Trace Status Report



The status code indicates the acceptance or the rejection of the subscriber trace at UP. Status code = 0 means, a success. Values 1-255 uniquely specifies the specific error code or notification. The list of error codes are defined post development.

Table 15: Error Code and Notification Table

Status Code	Status Description
MONSUB_SM_SUCCESS (0)	
MONSUB_SM_ERROR_FAILURE (1)	MonSub : Generic Failure status received
MONSUB_SM_ERROR_UNSUPPORTED (2)	MonSub : Unsupported Failure!
MONSUB_SM_ERROR_SESSION_EXIST_NONE (3)	MonSub : Session not Found!");
MONSUB_SM_ERROR_SESSION_LIMIT_EXCEED (4)	MonSub : Max Connections reached!
MONSUB_SM_ERROR_SESSION_INVALID_PARAM (5)	MonSub : Connect Message Failed!
MONSUB_SM_ERROR_SESSION_ALLOC_FAIL (6)	MonSub : Could not allocate monsub sesson at NPU!
MONSUB_SM_ERROR_CONFIG_INVALID_PARAM (7)	MonSub : Config Message Failed!
MONSUB_SM_ERROR_MONITOR_LIMIT_EXCEED (8)	MonSub : Max Stream Limit reached!
MONSUB_SM_ERROR_MONITOR_INVALID_PARAM (9)	MonSub : Monitor Message Failed!
MONSUB_SM_ERROR_MAX (10)	MonSub: Max Error!
MONSUB_COPROCDATA_CORRUPTED (11)	MonSub : File Handling Process Failed!
MONSUB_MAX_TRACING_SESSIONS_REACHED (12)	MonSub : Maximum Number of Tracing Sessions reached!

Status Code	Status Description
MONSUB_STOP_RECVD_WAIT_POLL_TIMEOUT (13)	MonSub : STOP notification is Successful. Wait till the poll-timeout configuration to start the next tracing!
MONSUB_FILECOPY_SOURCE_DIR_NOT_EXIST (14)	MonSub : Source Directory does not exist!
MONSUB_FILECOPY_DEST_DIR_NOT_EXIST (15)	MonSub : Destination Directory does not exist!
MONSUB_FILECOPY_SOURCE_DIR_OPEN_FAILURE (16)	MonSub : unable to open Source Directory!
MONSUB_FILECOPY_DEST_DIR_OPEN_FAILURE (17)	MonSub : Unable to open Destination Directory!
MONSUB_FILECOPY_SOURCE_OPEN_FAILED (18)	MonSub : Unable to open Source File!
MONSUB_FILECOPY_DESTINATION_OPEN_FAILED (19)	MonSub : Unable to open Destination File!
MONSUB_FILECOPY_DONE_FILE_DELETION_FAILED (20)	MonSub : Unable to delete .done file in Source Path!
MONSUB_FILECOPY_PCAP_FILE_DELETION_FAILED (21)	MonSub : Unable to delete .pcap file in Destination Path!
MONSUB_RESPONSE_NPUMGR_MONSUB_SESS_FAILED (22)	MonSub : Messenger Failure during Session Notification to NPUMGR!
MONSUB_RESPONSE_NPUMGR_MONSUB_CFG_FAILED (23)	MonSub-Config push to NPUMGR!
MONSUB_RESPONSE_NPUMGR_MONSUB_MONITOR_FAILED (24)	MonSub-Monitor Notification to NPUMGR!
MONSUB_RESPONSE_COPROC_FAILED (26)	MonSub : File Handling Process Failed!
MONSUB_RESPONSE_FILE_TRANSFER_SUCCESS (27)	MonSub: File Transfer successful.
MONSUB_RESPONSE_FILE_TRANSFER_FAILED (28)	MonSub: File Transfer failed!
MONSUB_ADMINISTRATIVE_DISCONNECT (29)	MonSub: Administrative Disconnect!
MONSUB_FILECOPY_DESTINATION_DISK_FULL (30)	MonSub: No space left in Destination Path!
MONSUB_FILECOPY_COPROC_ABRUPTLY_KILLED (31)	MonSub: File copy co-proc terminated abruptly!
MONSUB_LOGGING_COPROC_ABRUPTLY_KILLED (32)	MonSub: Logging co-proc terminated abruptly!
MONSUB_SM_DISCONNECT (33)	
MONSUB_FILECOPY_STATUS_MAX (34)	

Status Code	Status Description
Other	Internal Error adding protocol monitor trace - aborting…

Control Plane SMGR Functionality

Following are the modification in the CP SMGR to support this feature.

- Provide services to the CLI for enabling or disabling the MonSub tracing.
- When you enable the MonSub for a subscriber on Control Plane. The changes propagate to the corresponding U-Plane over Sx interface as per the instructions from CP CLI.
- Any tracing failures in the UP is reported to the CP (if MonSub enabled via CP console) by a "private IE Subscriber Trace Status Report" within Sx Session Report Request message from UP to CP.
- The feature supports the tracing of four concurrent subscriber tracing sessions for fast-path and slow-path PCAP creation from CP per User Plane instance.
- The CP instance sends the CLI instance ID while enabling MonSub from CP, so that the UP sends notifications to correct CP CLI instance ID.

Note There’s a race condition scenario when you enable the tracing for new/camp-on call. When the UE attach is in progress, private IE is sent in either Sx Establish Req or the Sx modify (existing attach sequence, so that the attach flow isn’t disturbed). For existing calls, the private IE is sent in the Sx modify request.

User Plane SMGR Functionality

Following are the modification in the UP SMGR to support this feature.

- Provide services to the CLI for enabling or disabling the MonSub tracing.
- Based on the MonSub private IE over Sx interface from C-PLANE. Enable or Disable the MonSub tracing and generate the ‘Subscriber Trace Status Report’ to inform C-PLANE whether tracing is on or not.
- Control NPUMGR to connect/start/stop/add/delete streams/tep bearers and disconnect.
- The SMGR maintains the PSN from the NPUMGR (as part of CONNECT API) and sub session id, which is SMGR (local to SMGR instance) specific. The SMGR sends all requests with PSN and sub session id to NPUMGR for a monitor subscriber tracing session.
- Based on the instructions from the CLI, configures panopticon (via NPUMGR) for changes such as packet size and priority.
- Read the ‘hex dump module’ configurations and store them locally. Pass the relevant parameters (such as filename) to Session Manager Co-Proc.

- Instantiate Session Manager Co-Proc and then instruct it to copy panopticon generated PCAP files to hard disk. Also handle the termination of Session Manager Co-Proc when MonSub session is over.
- Handle file copy message from Session Manager Co-Proc and inform panopticon about the copied bundle.
- If the file copy fails or there are problems with Session Manager Co-Proc instantiation, raises the SNMP alarms.
- Handle the buffer full indications from panopticon and copy the PCAP from the ram disk to the configured destination directory.
- Capture the control/slow-path packets. Pass them to Session Manager Co-Proc to publish it as a separate PCAP.
- This feature supports a maximum of four monitor subscriber tracing sessions for a U-PLANE instance. The NPUMGR enforces the tracing limit.
- The MonSub tracing session terminates in the absence of no space on hard disk or no hard disk.
- There are coproc (file copy and logging) per UP-SMGR instance, when monitor subscriber tracing is initiated for that SMGR instance.
- The MonSub session tear down takes time depending on the final poll timer and disconnect responses from co-proc/NPUMGR.

Note There is a race condition scenario while tracing is enabled for new/camp-on call. When the UE attach is in progress, private IE is sent in either Sx/N4 Establish Request or the Sx/N4 Modify Request (existing attach sequence, so that the attach flow is not disturbed). For existing calls, the private IE is sent in the Sx/N4 Modify Request.

Multi PDN Multi Trace

For a multi-PDN call, when you start the MonSub with Multi-trace=OFF, then it traces the only one PDN as a part of that MonSub session. When new PDN is initiated then existing PDN tracing stops and new PDN tracing starts. For this, first the new PDN tracing is started and then existing PDN tracing is STOPPED and hence new PSN and SMGR sub-session ID is allocated.

For a multi-PDN call, when you start the MonSub with Multi-trace=ON, then it traces the new PDN as a part of new FASTPATH tracing session (that is MonSub session). Hence after tracing the four PDN, MonSub CLI shows max tracing session reached. Tracing of the each PDN takes place as a separate MonSub session.

Note For Pure-S call, when MonSub starts from CP, then tracing of the multi-PDN happens as a separate FASTPATH tracing session (that is separate MonSub session) irrespective of MT=ON or OFF.

MonSub Stats

A new mechanism is added to publish the stats regarding the quality of FASTPATH PCAP capture on MonSub CLI. The new mechanism publishes the stats whenever it receives the buffer full MEH indication at SESSMGR, throttled at every five seconds. The feature supports a maximum of four buffers for a FASTPATH PCAP corresponding to MonSub session. The feature does not publish the stats by default and needs to be enabled via debug CLI on UP.

- **debug uplane monsub-stats disabled**
- **debug uplane monsub-stats enabled**

The stats contains the following informations:

```
Packet accepted: 14250000                          Packet rejected: 62297
Congestion Short Term: 0                            Congestion Longer Term: 0
Throttled: 0                                                  PCAP File Transfer Rate: 9.91
 mbps
```

The PCAP file transfer rate is the rate at which copy co-proc writes the PCAP from RAM-FS to HD-RAID.

X-Header

This feature supports the X-Header capture in slow-path PCAP. The PGW-U inserts the X-HEADER for Uplink packet. The PGW-U captures the packet at entry and exit interfaces. So, the exit packet sent to SGi contains the inserted x-header.

The PGW-U inserts the X-HEADER for Downlink packet. The PGW-U captures the packet at entry and exit interfaces. So, the exit packet sent to S5-U or S1-U contains the inserted x-header.

How It Works

The Monitor Subscriber feature is discussed in detail in the following sections:

Configuration Procedure for Monitor Subscriber

The protocol monitor can be used to display information for a specific subscriber session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Follow the instructions in this section to invoke and configure the protocol monitoring tool for a specific subscriber session.

Step 1 Invoke the monitor subscriber command from the Exec mode by entering the **monitor subscriber** CLI command.

```
[local]host_name# monitor subscriber { callid | imei | imsi | ipaddr | ipv6addr |
msid | msisdn | next-call | pcf | peer-fa | peer-lac | sgsn-address | type |
username }
```

An output listing all the currently available protocols, each with an assigned number, is displayed. Specify the method the monitor should use by entering the appropriate keyword.

Step 2 Specify the method the monitor should use by entering the appropriate keyword.

Select other options and/or enter the appropriate information for the selected keyword.

Step 3 Select other options and/or enter the appropriate information for the selected keyword.

If no session matching the specified criteria was being processed when the monitor was invoked, a screen of available monitoring options appears.

Step 4 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter or 2-digit number associated with that option (C, D, E, 11, 12, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Option **Y** for performing multi-call traces is only supported for use with the GGSN.

```
WARNING!!! You have selected options that can DISRUPT USER SERVICE
Existing CALLS MAY BE DROPPED and/or new CALLS MAY FAIL!!!
(Under heavy call load, some debugging output may not be displayed)
Proceed? - Select (Y)es or (N)o
```

Step 5 Repeat step 6 as needed to enable or disable multiple protocols.

Step 6 Press the **Enter** key to refresh the screen and begin monitoring.

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Monsub CLI Options

The following options with their default value are added to existing **monitor subscriber** command.

UPF Monitor Subscriber CLI

Following are the options:

- **W** - UP PCAP Trace (ON): This parameter is used to create PCAP trace for slowpath and fastpath.
- **U** - Mon Display (ON): The non-protocol events (such as statistics and charging information from ECS and so on) are also captured in slowpath PCAP files and are displayed on UPF monitor console.
- **V** - PCAP Hexdump (ON): This flag must be set to ON to capture the protocol packets in a text file in hexdump format on UPF.

Note Currently, UP PCAP Trace flag must be set to ON to capture fastpath and slowpath PCAP files.

- **F - Packet Capture (Full Pkt)**: Captures all packets from fastpath.

  Using this option, operators can choose between full and partial packet captures. By entering **F**, the packet capture type can be changed to either full or partial. With partial packet capture, users can enter

packet sizes from 1 to 16384 bytes. For example, if input is given as 20, only the first 20 bytes of fastpath packets will be captured and the remaining packets will be dropped.

Note When opening the PCAP file, the summary view will display full length of the packet, but the detailed view will show only the truncated packet.

- **/ - Priority (0)**: The value is in the range from "0 – Best Effort" to "7 – Guaranteed"
  - 0 - Best Effort
  - 1 - Low
  - 2 - Med-Low
  - 3 - Medium
  - 4 - Med-High
  - 5 - High
  - 6 - Critical
  - 7 - Guaranteed

Caution It is strongly recommended to not change the default value. It can adversely affect the system performance.

- **N - MEH Header (OFF)** : The MEH header is stripped from the IP packet if this option is configured

Show Monitor Subscriber Sessions

Following is the new CLI to show the ongoing MonSub session.

You can trigger the **show monitor subscriber fastpath session all** CLI command from both SMF and UPF. You can trigger the **show monitor subscriber fastpath session up-ip-address** CLI command from the SMF

- **SessId:** This is the local session id for MonSub session on UPF Sessmgr.
- **CallID:** Call ID on UPF.
- **PSN:** This is panopticon sequence number. There is a maximum of four MonSub fastpath tracing sessions on one UPF with PSN ranging from 0-3.
- **Start time:** Time at which MonSub tracing session starts.
- **Interface Type:** This is to identify the call type for which MonSub fastpath tracing session was started, whether it is Sxa, Sxb or Sxab.

Disconnect Monitor Subscriber Sessions

Following is the new CLI to disconnect the ongoing MonSub session. You can trigger the CLI from both CP and UP.

monitor subscriber fastpath disconnect sessmgr-instance *upf_sessmgr_instance_id* **session-id** *local_monsub_sessid_sessmgr_level*

If the MonSub session disconnect is successful, the following message dispalys on console.

```
Session Disconnected Successfully
```

If the MonSub session disconnect fails, the following message dispalys on console.

```
Monitor Subscriber session does not exist
```

Note Only security administrator can execute the monitor disconnect CLI.

Context, CDRMOD and Hexdump Interaction for Monitor Subscriber

Hexdump module must be configured to provide operators the provision to configure Files names and Poll timers. The Hexdump module is one of the modules such as—EDR, UDR and so on, that are part of the CDRMOD functionality. It is recommended to configure the hexdump in a non-local context such as the ECS context. Hexdump modules are not supported in local context.

For more information on Hexdump module and its configuration, refer to the *Packet Capture (PCAP) Trace* chapter in the *ASR5500 System Administration Guide*

PCAP File Name Convention

The naming conventions for PCAP files are discussed in the following sections

Note Only **monitor-subscriber-file-name** and **rotation** options are used in naming PCAP files.

Slowpath File Name Convention

The slow path file names appear in the following format:

```
curr_slowpath_{SMGR Mon Sub Session
Id}_{monsub_file_name_option_val}_{Timestamp}_{RotationCount}.pcap
```

or

```
slowpath_{SMGR Mon Sub Session
Id}_{monsub_file_name_option_val}_{Timestamp}_{RotationCount}.pcap
```

File with ‘curr_’ prefix is the file, that is currently being written to, that is still not closed. When files are to be rotated (depending on the file rotation parameters), file without the ‘curr_’ prefix are copied to hard disk.

The SMGR MonSub Session Id – This is the session Id for MonSub session created on Uplane SMGR instance ID, which created this PCAP. This Id is local to SMGR instance, so there could be two SLOWPATH pcap captured with same ID.

When files are to be copied to hard disk, The monsub_file_name_option_val is replaced by:

- IMSI value if **monitor-subscriber-file-name** is set to "imsi".
- Call ID value if **monitor-subscriber-file-name** is set to "call-id"

- Username value if **monitor-subscriber-file-name** is set to ‘username’

Timestamp is in the following format "MMDDYYYYHHMMSS", where:

- MM - Month, DD - Date and YYYY - Year.
- HH -Hour, MM - Minutes and SS - Seconds.

RotationCount is a 9-digit value that is incremented every time an old file is rotated, and a new file is generated.

00000000 for the first file, 00000001 for the second file and so on.

Rotation of slowpath files is determined by following option in **hexdump-module file** configuration:

rotation { num-records *number* **| time** *seconds*| **volume** *bytes* **}**

- **num-records:** num-records specifies the number of packets after which a new file is generated and ‘RotationCount’ in the filename is incremented. The range of number is between 100 to 10240, and the default value is 1024.
- **time:** time specifies the time to wait in seconds before a new file is generated and ‘RotationCount’ in the filename is incremented. seconds must be an integer from 30 through 86400. The default value is 3600.
- **volume:** volume specifies the number of bytes after which a new file is generated and ‘RotationCount’ in the filename is incremented. bytes must be an integer from 51200 through 62914560. The default value is 102400.

Note The **tarriff-time** parameter under rotation is ignored as it is not suitable for PCAP file capture.

The following are examples of the file naming conventions for slowpath PCAP files.

- For the ‘imsi’ option where IMSI is ‘112233445566778’, slowpath files are named as:

  ```
  slowpath_S0_112233445566778_07152019050907_000000000.pcap
  ```

- For ‘call_id’ option where Call Id is ‘01317b22’, slowpath files are named as:

  ```
  slowpath_S0_01317b22_07152019050907_000000000.pcap
  ```

Note The parameter **tarrif-time** is not applicable for PCAP file capture.

Fastpath File Name Convention

The fastpath file names appear in the following format:

```
vpp_{S}_{B}_{encap}_{monsub_file_name_option}_{Timestamp}_{FileCount}.pcap
```

- S is replaced by either ‘S1’, ‘S2’,'S3’, or 'S4’.
- B is replaced by either ‘B0’,‘B1’,'B2', or 'B3' depending on the bundle generated by Panopticon.
- monsub_file_name_option is replaced by:
  - IMSI value if **monitor-subscriber-file-name** is set to "imsi".

- Call ID value if **monitor-subscriber-file-name** is set to "call-id"
- Username value if **monitor-subscriber-file-name** is set to ‘username’

Timestamp is in the following format "MMDDYYYYHHMMSS", where:

- MM - Month, DD - Date and YYYY - Year.
- HH -Hour, MM - Minutes and SS - Seconds.

RotationCount is a 9-digit value that is incremented every time an old file is rotated, and a new file is generated.

00000000 for the first file, 00000001 for the second file and so on.

Fast path "FileCount" is not the same as the slowpath "RotationCount" parameters and hence ‘hexdump-module file rotation’ parameters are ignored while naming fastpath files.

In Phases 1 of the feature, fastpath generated file names are like ‘vpp_S1_B0_ip.pcap’ or ‘vpp_S1_B1_ip.pcap’, they are renamed to following when being copied over to non-volatile storage:

- `vpp_S1_B0_ip_01317b22_07152019050907_000000000.pcap`
- `vpp_S1_B1_ip_01317b22_07152019050908_000000001.pcap`
- `vpp_S1_B0_ip_01317b22_07152019050908_000000002.pcap`

In MonSub phase 3, a PCAP “bundle” is replaced with a single PCAP file that uses Ethernet encapsulation.

In Phase 3, each fastpath session file is captured in the Ethernet PCAP file that is ‘vpp_S0_B0_eth.pcap’ and they are renamed to following when being copied to a non-volatile storage:

`vpp_S0_B0_eth_01317b22_07152019050907_000000000.pcap`

For ‘callid’ option where Call Id is ‘12345678ef’:

- `slowpath_S0_12345678ef _07152019050907_000000000.pcap`
- `vpp_S1_B0_eth_12345678ef _07152019050907_000000000.pcap`

For ‘username’ option where username is ‘9890098900’:

- `slowpath_S0_07152019050907_000000000_9890098900.pcap`
- `vpp_S1_B0_eth_07152019050907_000000000_9890098900.pcap`

PCAP File Location

Fastpath PCAP files are written to the `/records/pcap` directory in same card and CPU complex where the SMGR owns the subscriber session resides.

`/records` directory is mapped to the "tmpfs" filesystem that is mapped to RAM. In this state, the files are suffixed with a ".pending" extension. For example:

```
-rw-rw-r-- 1 root root 268599296 Sep 23 14:04 vpp_S1_B0_eth.pending
```

Note The files size at this stage is not the actual file size when it is written to a persistent storage.

Once the fastpath tracing mechanism has written the files, they are converted to '.pcap' files and renamed as given below. Additionally, there is a file that ends with a ".done" extension:

```
-rw-rw-r--    1 root     root       8689188 Oct 16 22:06 vpp_S0_B0_eth.pcap
```

After the PCAP files are written by fastpath tracing mechanism, the Co-Proc functionality instantiates and copies the files to a hard disk or a persistent storage.

The above file location process for Fastpath is also applicable to Slowpath.

The target file location in all cases is: `/hd_raid/records/hexdump`, except for the case in the hexdump module configuration where **use-harddisk** is enabled and the **directory** option under the **hexdump file** is to a custom value. For example, if the **directory** option is set to a value "abc" then the target location for the PCAP file will be: `/hd_raid/records/hexdump/abc/`.

In this feature implementation, a predefined location is set for PCAP files.

- To make sure that `/records/pcap` directory is not populated when issues are encountered with the use of **use-harddisk** and **hexdump module** configurations.
- For regular cleanup from `/hd_raid/records/hexdump` directory.

File Transfer to External Location

Once the files have been copied to the hard disk, they can be copied over to an external server using the command: **transfer-mode** option under the **hexdump** command in the **hexdump-module** configuration.

Apart from **transfer-mode**, other relevant options under **hexdump** can be used for external file transfer. Operators can use these commands to avoid excessive storage during fastpath processing.

Limitations

Following are the Limitations:

- Restarting trace immediately after quitting may result in fastpath files in `/records/pcap` directory to be overwritten. It is recommended to restart the session after a brief moment (a few seconds).
- When MonSub trace is stopped, the tear down process can take a few seconds, so it is recommended to wait for few seconds. A maximum of five seconds (hexdump poll timer value in seconds) before toggling the MonSub trace to start, else, operators may observe MAX TRACING SESSIONS REACHED momentarily.
- Show monitor subscriber fastpath sessions CLI does not display the MonSub sessions that are being stopped. Hence there is a transient period where new MonSub sessions can be rejected due to max sessions reached, whereas show CLI shows less sessions. It is recommended that operators wait for some time before starting a new MonSub trace session.
- Changing fastpath configuration options is only possible when **UP Pcap Trace** is set to OFF.
- When MT=ON in the Multi-PDN, then once MT=OFF, new PDN tracing is not started due to MAX TRACING REACHED, and then all other tracing is STOPPED. This is because the first new PDN tracing is started and then all previous PDN's were STOPPED for MT=OFF case.
- It is recommended to not to launch the same UE MonSub sessions from different CLIs.

- In slowpath PCAP, the egress DL packets does not show the GTPU-U header because the functionality to add GTP-U is with fastpath. So, ingress and egress DL packets shows up the duplicates, unless there is some packet modification like HTTP X-headers applied over the ingress packets.
- Toggling C and D options does not impact the PCAP capture.
- For Multi-PDN, the fastpath filenames does not use the Call Id, because, by definition the multi-PDN case has more than one call id and hence a higher-level configuration such as IMSI is more suitable for naming the files.
- Only the named options explicitly mentioned in this document are supported from *hexdump-module file* configuration.
- Number of streams that can be traced in fastpath is limited to 5000. Stream is defined as a TCP or UDP flow which is made up of (source IP address, destination IP address, source port, and destination port, transport protocol (TCP or UDP)).
- Fastpath packets cannot be streamed to an external server. They are stored on the hard-disk and transferred (either manually) or by using **transfer-mode** options.
- The UP PCAP trace must be set to ON to capture fastpath and slowpath PCAP files.
- MonSub CLI option '<SPACE> Pause' is only to pause console events. There is no impact on other tracing events (slowpath PCAP, fastpath PCAP and protocol packets tracing in a text file in hexdump format) with this option.
- The UP trace PCAP file does not contain the initial PFCP Sx Request/Response, due to race condition.
- The ICMP Packets and a first packets of TCP and UDP streams flow through both slowpath and fastpath. Default values of GTPU (option 26) and User L3 (Option 19) are set to OFF. As a result, these packets are not captured in slowpath captures. If Option 26 is set to ON then these packets are captured in slowpath PCAP captures. As mentioned in previous point, option 19 has no effect on slowpath PCAP capture.
- Data Events flag must be set to ON to capture fastpath and slowpath PCAP files.
- Only firstPDN tracing is supported forPure-Scall. This limitation will be fixed with multiple subscriber tracing support.
- The Mon sub tracing is not supported for option **Next-SAEGW Call** on UP.
- The Mon sub tracing is not supported for option **Next call by APN** for Pure-S call type.
- On ASR 5500 setup with the default value of poll-timer, all the packets may not be captured due to a known issue. To avoid large number of packets to be rejected, it is recommended to change the poll-timer value to the lowest possible (10ms).
- If context replacement occurs (if the same subscriber reattaches without a detach) then the slowpath captures for the new call continues to be in the old slowpath files.

Configuring the Hexdump Module for MonSub in UPF

Configuring MonSub Poll Timer

Use this configuration to set the frequency of PCAP file capture check.

```
configure
  context context_name
    hexdump-module
      hexdump monitor-subscriber-poll-timeout poll_timer_value
      end
```

NOTES:

- **hexdump monitor-subscriber-poll-timeout** : This option specifies how frequently the check for newly captured PCAP files in the volatile storage must be done before they are copied to persistent storage.
- *poll_timer_value*: Specifies the poll timer value in milliseconds. It must be an integer in the range of 10 ms to 60 seconds. Default: 30 seconds.

Note It is strongly recommended to not configure the timer with a value less than 5 seconds.

- This option is only applicable when MonSub is enabled for the products that have fastpath functionality - PGW, SAEGW on ASR-5500 and VPC-SI.

Configuring MonSub File Name

Use the following configuration to specify the file name of the PCAP file which contains IMSI, Call ID or Username.

```
configure
  context context_name
    hexdump-module
      file rotation { num-records number | tariff-time minute minutes hour
 hours | time seconds| volume bytes | monitor-subscriber-file-name { imsi |
username | call-id }
      end
```

NOTES:

- **monitor-subscriber-file-name { imsi | username | call-id }**: This option specifies if the name of the captured PCAP files will contain IMSI, Call Id or Username. This option is only applicable on products that have fastpath functionality (PGW, SAEGW on ASR 5500 and VPC-SI) AND only when Monitor Subscriber functionality is enabled. Default: IMSI.
- **rotation { num-records** *number* **| tariff-time minute** *minutes* **hour** *hours* **| time** *seconds* **| volume** *bytes* **}**: Specifies when to close a hexdump file and create a new one.
  - **num-records** *number* : Specifies the maximum number of records that should be added to a hexdump file. When the number of records in the file reaches this value, the file is complete.

    number must be an integer from 100 through 10240. Default: 1024
  - **tariff-time minute** *minutes* **hour** *hours* : Specifies to close the current hexdump file and create a new one based on the tariff time (in minutes and hours).

    minutes must be an integer from 0 through 59.

    hours must be an integer from 0 through 23.

- **time** *seconds* : Specifies the period of time to wait (in seconds) before closing the current hexdump file and creating a new one.

  seconds must be an integer from 30 through 86400. Default: 3600

Important It is recommended to set the rotation time to 30 seconds.

- **volume** *bytes* : Specifies the maximum size of the hexdump file (in bytes) before closing it and creating a new one.

  bytes must be an integer from 51200 through 62914560. Note that a higher setting may improve the compression ratio when the compression keyword is set to gzip. Default: 102400

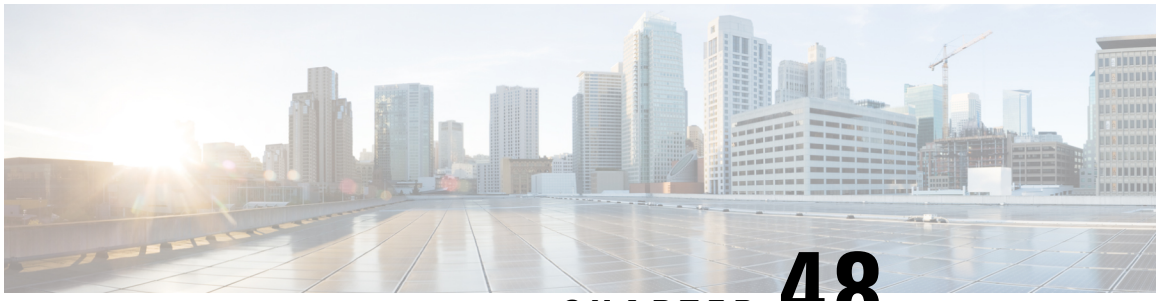
Monitoring and Troubleshooting

This section provides information regarding monitoring and troubleshooting the Monitor Subscriber feature.

SNMP Traps

The following SNMP trap(s) are added in support of the Monitor Subscriber feature:

- MonSubProcessInitFailure: This trap is triggered when MonSub handler process has failed for a particular process and service.



CHAPTER 48

MPLS Support on VPC-SI for CUPS

- Revision History, on page 375
- Feature Description, on page 375
- How it Works, on page 376
- Monitoring and Troubleshooting, on page 387

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In the existing platforms (VPC-DI, ASR 5500), the boxer supports MPLS, which uses the underlying dataplane forwarder to switch MPLS traffic. In ASR 5500, the NP4c network processor generates and processes MPLS traffic while in VPC-DI, the IFTask generates and processes MPLS traffic.

The MPLS Support on VPC-SI for CUPS feature enables MPLS support on VPC-SI (SI-CUPS), which uses VPP as the dataplane forwarder.

VPP supports and provides multiple dataplane features that include the MPLS stack as a separate graph node. VPP generates labeled packets and simultaneously processes incoming labeled packets. This helps differentiate between different customer VRFs to support a large number of corporate APNs having different addressing models and requirements.

The MPLS Support on VPC-SI for CUPS feature supports the following functionalities:

- Uses the VPP MPLS stack to send the MPLS labeled packet.
- Uses the VPP MPLS stack to process the incoming labeled MPLS packet.

- Supports all existing MPLS configuration (VPC-DI, ASR 5500) and provides feature parity with new deployments using VPC-SI CUPS.
- Supports VPPCTL CLI commands to display NHLFE and ILM tables that are in VPP for debugging and comparing values with boxer configuration.

How it Works

This section briefly describes how the MPLS Support on VPC-SI for CUPS works.

In the current CUPs architecture, VPP forwarder provides its own MPLS stack, which supports all the existing functionalities for MPLS packet processing. The VPP MPLS stack is configured with the appropriate Next-Hop Label Forwarding Entry (NHLFE) and incoming label map (ILM) tables. This helps generate the MPLS packet on the egress with the correct MPLS header. It also processes the incoming MPLS packet and switches this packet based on the incoming labels to the appropriate next hop table identifier (VRF context of the subscriber) based on the incoming label.

The MPLS solution supports the following scenarios:

- MPLS-CE Connected to PE
- VPC-SI as a PE

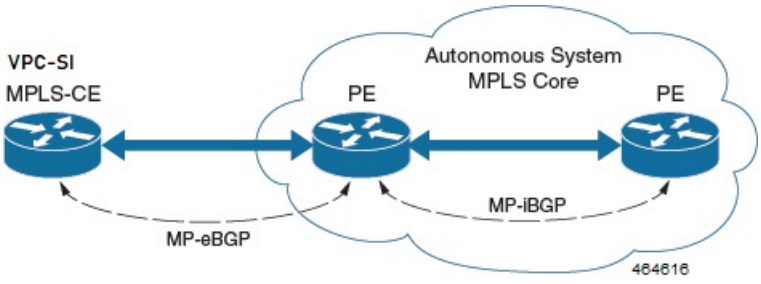
VPC-SI also supports VPNv6 as described in RFC 4659 – *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*.

MPLS-CE Connected to PE

The VPC-SI functions as an MPLS-CE (Customer Edge) network element connected to a Provider Edge (PE) Label Edge Router (LER), which in turn connects to the MPLS core as per RFC 4364.

The following figure illustrates the MPLS-CE to PE connection:

Figure 15: VPC-SI MPLS-CE to PE



The MPLS-CE functions like a PE router within its own Autonomous System (AS). It maintains Virtual Routing and Forwarding (VRF) routes and exchanges VPN route information with the PE through an MP-eBGP (Multi Protocol external BGP) session.

The PE is also configured with VRFs and exchanges VPN routes with other PEs in its AS through MP-iBGP (Multi Protocol internal BGP) connection and MPLS-CE through an MP-eBGP connection.

The EBGP connection allows the PE to change next-hop IP addresses and labels in the routes learnt from IBGP peers before advertising them to the MPLS-CE. The MPLS-CE uses only MP-eBGP to advertise and

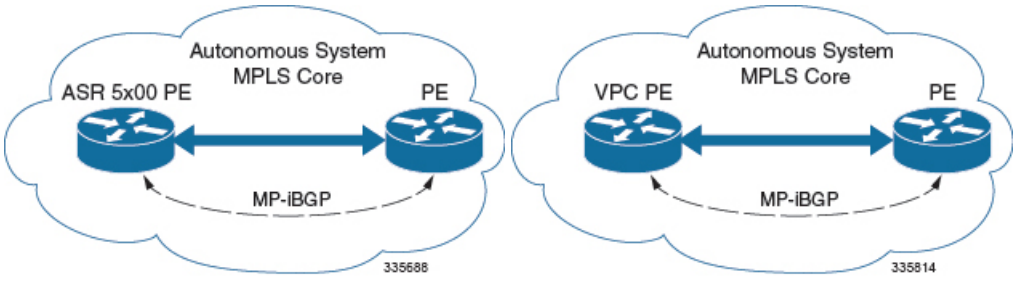
learn routes. Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) are not required because of direct-connect EBGP peering. The MPLS-CE pushes or pops a single label (learnt over the MP-eBGP connection) to or from the PE.

VPC-SI as a PE

Overview

In this scenario, the VPC-SI functions as a PE router sitting at the edge of the MPLS core. See the figure below.

Figure 16: VPC-SI as a PE



The VPC-SI eliminates the need for an ASBR or PE as shown in the first two scenarios. In this scenario, two main requirements are introduced: IBGP functionality and MPLS label distribution protocols.

The VPC-SI can be configured to add two labels:

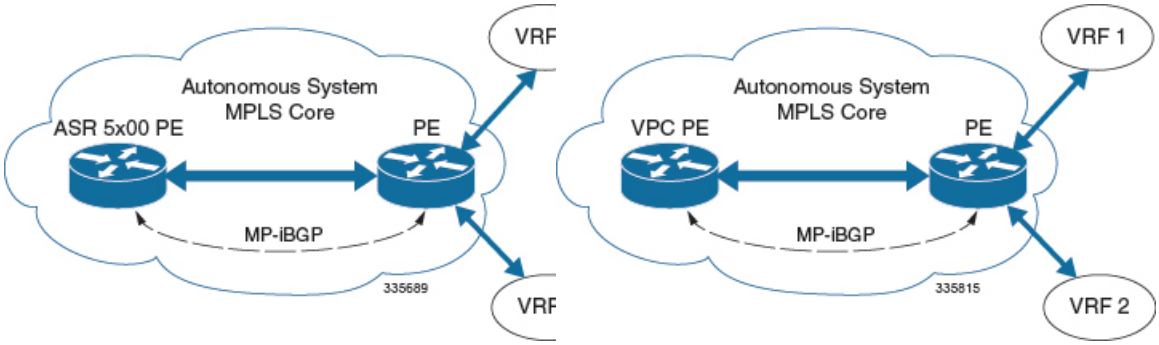
- an outer label learned from LDP or RSVP-TE (RSVP-Traffic Engineering)
- an inner label learned from MP-iBGP

This solution supports traffic engineering and QoS initiated via the VPC-SI.

Sample Configuration

In this example, VRFs are configured on the ASR 5500 PE and pools are associated with VRFs. The VPC-SI exchanges VPN routes with its IBGP peers (PE routers) and learns the MPLS paths to reach PEs via LDP. The VPC-SI forwards the packets to the next-hop with two labels – an inner label learned from PE and an outer label learned from the next hop IBGP neighbor.

Figure 17: Sample Configuration



```
mpls ip
  protocol ldp
    enable
  exit
exit

ip vrf vrf1
  mpls traffic-class copy
exit
ip vrf vrf2
  mpls traffic-class value 5
exit

router bgp 300
  ip vrf vrf1
    route-target export 300 1
      route-target import 300 1
      route-distinguisher 300 1
  exit
    ip vrf vrf2
      route-target export 300 2
        route-target import 300 2
        route-distinguisher 300 2
  exit

  router-id 209.165.201.1
  neighbor 209.165.200.225 remote-as 300
  neighbor 209.165.200.225 update-source node1_loopback

  address-family vpnv4
    neighbor 209.165.200.225 activate
      neighbor 209.165.200.225 send-community both
      neighbor 209.165.200.225 next-hop-self
  exit

  address-family ipv4 vrf vrf1
    redistribute connected
  exit

  address-family ipv4 vrf vrf2
    redistribute connected
exit

interface interface_to_internet
  ip address 209.165.200.224/27
  mpls ip
exit
router ospf
  network 209.165.201.0/27 area 209.165.201.5
exit
```

IPv6 Support for BGP MPLS VPNs

Overview

The VPC-SI supports VPNv6 as described in RFC 4659 – *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*.

An IPv6 VPN is connected over an IPv6 interface or sub-interface to the Service Provider (SP) backbone via a PE router. The site can be both IPv4 and IPv6 capable. Each VPNv6 has its own address space which means

a given address denotes different systems in different VPNs. This is achieved via a VPNv6 address-family which prepends a Route Distinguisher (RD) to the IP address.

A VPNv6 address is a 24-byte quantity beginning with an 8-byte RD and ending with a 16-byte IPv6 address. When a site is IPv4 and IPv6 capable, the same RD can be used for the advertisement of both IPv4 and IPv6 addresses.

The system appends RD to IPv6 routes and exchanges the labeled IPv6-RD using the VPNv6 address-family. The Address Family Identifier (AFI) and Subsequent Address Family Identifier (SAFI) fields for VPNv6 routes will be set to 2 and 128 respectively.

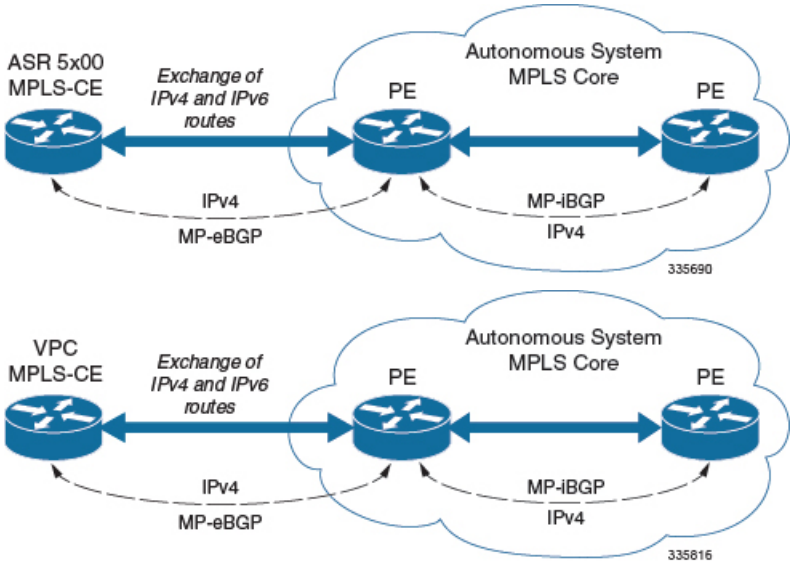
The IPv6 VPN traffic will be transported to the BGP speaker via IPv4 tunneling. The BGP speaker advertises to its peer a Next Hop Network Address field containing a VPN-IPv6 address whose 8-octet RD is set to zero and whose 16-octet IPv6 address is encoded as an IPv4-mapped IPv6 address (RFC 4291) containing the IPv4 address of the advertising router. It is assumed that only EBGP peering will be used to exchange VPNv6 routes.

Support for VPN-IPv6 assumes the following:

- Dual Stack (IPv4/IPv6) routing
- IPv6 pools in VRFs
- BGP peering over a directly connected IPv4 interface

See the figure below.

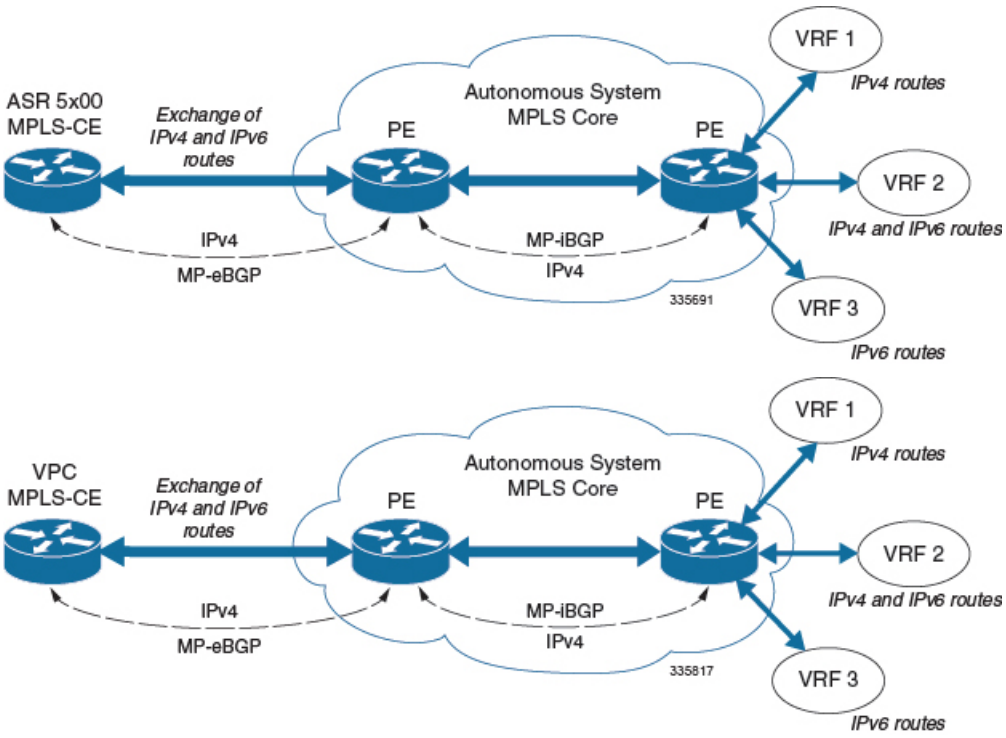
Figure 18: IPv6-RD Support for VPNv6



Sample Configuration

This example assumes three VRFs. VRF 1 has only IPv4 routes, VRF 2 has both IPv4 and IPv6 routes, and VRF 3 has only IPv6 routes.

Figure 19: VPNv6 Sample Configuration



Configure VRFs.

```
  ip vrf vrf1
  exit
  ip vrf vrf2
  exit
  ip vrf vrf3
  exit
```

Enable MPLS BGP forwarding.

```
  mpls bgp forwarding
```

Configure pools.

```
  ip pool vrf1-pool 209.165.200.230 255.255.255.224 private 0 vrf vrf1
  exit
  ip pool vrf2-pool 209.165.200.230 255.255.255.224 private 0 vrf vrf2

  exit
  ipv6 pool vrf2-v6pool prefix 2005:0101::/32 private 0 vrf vrf2
  exit
  ipv6 pool vrf3-v6pool prefix 2005:0101::/32 private 0 vrf vrf3
exit
```

Configure interfaces.

```
  interface ce_interface_to_rtr
    ip address 209.165.200.226 255.255.255.224
  exit
  interface ce_v6_interface
    ip address 2009:0101:0101:0101::1/96
  exit
  interface ce_loopback loopback
    ip address 209.165.200.227 255.255.255.255
```

```
exit
interface vrf1-loop loopback
  ip vrf forwarding vrf1
  ip address 209.165.200.228 255.255.255.255
exit
interface vrf2-loop loopback
  ip vrf forwarding vrf2
  ip address 209.165.200.229 255.255.255.255
exit
interface vrf2-v6loop loopback
  ip vrf forwarding vrf2
  ip address 2005:0202:0101::1/128
exit
interface vrf3-v6loop loopback
  ip vrf forwarding vrf3
  ip address 2005:0303:0101::1/128
exit
```

Configure BGP along with address families and redistribution rules.

```
router bgp 800
  router-id 209.165.200.225
neighbor 209.165.200.240 remote-as 1003
  neighbor 209.165.200.240 activate
address-family vpnv4
  neighbor 209.165.200.240 activate
  neighbor 209.165.200.240 send-community both
exit
address-family vpnv6
  neighbor 209.165.200.240 activate
  neighbor 209.165.200.240 send-community both
exit
ip vrf vrf1
  route-distinguisher 800 1
  route-target export 800 1
  route-target import 800 1
exit
address-family ipv4 vrf vrf1
  redistribute connected
  redistribute static
exit
ip vrf vrf2
  route-distinguisher 800 2
  route-target export 800 2
  route-target import 800 2
exit
address-family ipv4 vrf vrf2
  redistribute connected
  redistribute static
exit
address-family ipv6 vrf vrf2
  redistribute connected
  redistribute static
exit
ip vrf vrf3
  route-distinguisher 800 3
  route-target export 800 3
  route-target import 800 3
exit
address-family ipv6 vrf   vrf3
  redistribute connected
  redistribute static
exit
```

Configure APNs.

```
apn walmart51.com
  selection-mode sent-by-ms
  accounting-mode none
  aaa group walmart-group
  authentication pap 1 chap 2 allow-noauth
  ip context-name Gi_ce
  ip address pool name vrf1-pool
exit
apn amazon51.com
  selection-mode sent-by-ms
  accounting-mode none
  aaa group amazon-group
  authentication pap 1 chap 2 allow-noauth
  ip context-name Gi_ce
  ip address pool name vrf2-pool
  ipv6 address prefix-pool vrf2-v6pool
exit
apn apple51.com
  selection-mode sent-by-ms
  accounting-mode none
  aaa group apple-group
  authentication pap 1 chap 2 allow-noauthip context-name Gi_ce
  ipv6 address prefix-pool vrf3-v6pool
exit
aaa-group amazon-group
  radius ip vrf vrf2
aaa group default
exit
gtpp group default
exit
ip igmp profile default
exit
```

Bind physical interfaces with the port.

VPN-Related CLI Commands

VPN-related features and functions are supported across several CLI command modes. The following tables identify commands associated with configuration and monitoring of VPN-related functions.

Table 16: VPN-Related Configuration Commands

CLI Mode	Command	Description
BGP Address-Family (IPv4/IPv6) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.
BGP Address-Family (IPv4/IPv6) Configuration Mode	**neighbor** *ip_address* **send community { both | extended | standard }**	Sends the community attributes to a peer router (neighbor).
BGP Address-Family (IPv4/IPv6) Configuration Mode	**redistribute connected**	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Address-Family (VPNv4) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.

CLI Mode	Command	Description
BGP Address-Family (VPNv4) Configuration Mode	**neighbor** *ip_address* **send community { both | extended | standard }**	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.
BGP Address-Family (VRF) Configuration Mode	**neighbor** *ip_address* **send community { both | extended | standard }**	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	**redistribute connected**	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Configuration Mode	**address-family { ipv4 vrf** *vrf_name* **| vpnv4 }**	Enables the exchange of IPv4 VRF routing information. There is a different mode for each address-family.
BGP Configuration Mode	**address-family { ipv6 vrf** *vrf_name* | **vpnv6 }**	Configures a VPNv6 address family and IPv6 VRF routing in BGP.
BGP Configuration Mode	**ip vrf** *vrf_name*	Adds a VRF to BGP and switches to the VRF Configuration mode to allow configuration of BGP attributes for the VRF.
BGP IP VRF Configuration Mode	**route-distinguisher {** *as_value* | *ip_address* **}** *rd_value*	Assigns a Route Distinguisher (RD) for the VRF. The RD value must be a unique value on the router for each VRF.
BGP IP VRF Configuration Mode	**route-target { both | import | export } {** *as_value* | *ip_address* **}** *rt_value*	Adds a list of import and export route-target extended communities to the VRF.

CLI Mode	Command	Description
Context Configuration Mode	**ip pool** *pool_name addr_range* **vrf** *vrf_name* **[mpls-label input** *inlabel1* **output** *outlabel1 outlabel2* **]**	Configures a pool into the specified VRF. This parameter must be specified with the Next-Hop parameter. *inlabel1* is the MPLS label that identifies inbound traffic destined for this pool. *outlabel1* and *outlabel2* specify the MPLS labels to be added to packets sent for subscribers from this pool.
Context Configuration Mode	**ip vrf** *vrf_name*	Creates a VRF and assigns a VRF-ID. A VRF is created in the router.
Context Configuration Mode	**ipv6 pool** *pool_name* **vrf** *vrf_name*	Associates the pool with that VRF. **Note:** By default the configured ipv6 pool will be associated with the global routing domain.
Context Configuration Mode	**mpls bgp forwarding**	Globally enables MPLS Border Gateway Protocol (BGP) forwarding.
Context Configuration Mode	**mpls exp** *value*	Sets the default behavior as Best Effort using a zero value in the 3-bit MPLS EXP header. This value applies to all the VRFs in the context. The default behavior is to copy the DSCP value of mobile subscriber traffic to the EXP header, if there is no explicit configuration for DSCP to EXP (via the **mpls map-dscp-to-exp dscp** *n* **exp** *m* command). **mpls exp** disables the default behavior and sets the EXP value to the configured *value*.
Context Configuration Mode	**mpls ip**	Globally enables the MPLS forwarding of IPv4 packets along normally routed paths.
Context Configuration Mode	**radius change-authorize-nas-ip ip_address** *ip_address* **{ encrypted | key }** *value* **port** *port_num* **mpls input** *inlabel* **output** *outlabel1 outlabel2*	Configures COA traffic to use the specified MPLS labels. *inlabel* identifies inbound COA traffic. *outlabel1* and *outlabel2* specify the MPLS labels to be added to the COA response. *outlabel1* is the inner output label; *outlabel2* is the outer output label.

CLI Mode	Command	Description
Ethernet Interface Configuration Mode	**mpls ip**	Enables dynamic MPLS forwarding of IP packets on this interface.
Exec Mode	**clear ip bgp peer**	Clears BGP sessions.
Exec Mode	**lsp-ping** *ip_prefix_FEC*	Checks MPLS Label-Switched Path (LSP) connectivity for the specified forwarding equivalence class (FEC). It must be followed by an IPv4 or IPv6 FEC prefix.
Exec Mode	**lsp-traceroute** *ip_prefix_FEC*	Discovers MPLS LSP routes that packets actually take when traveling to their destinations. It must be followed by an IPv4 or IPv6 FEC prefix.
IP VRF Context Configuration Mode	**mpls map-dscp-to-exp dscp** *dscp_bit_value* **exp** *exp_bit_value*	Maps the final differentiated services code point (DSCP) bit value in the IP packet header to the final Experimental (EXP) bit value in the MPLS header for incoming traffic.
IP VRF Context Configuration Mode	**mpls map-exp-to-dscp exp** *exp_bit_value* **dscp** *dscp_bit_value*	Maps the incoming EXP bit value in the MPLS header to the internal DSCP bit value in IP packet headers for outgoing traffic.
MPLS-IP Configuration Mode	**protocol ldp**	Creates the MPLS protocol family configuration modes, or configures an existing protocol and enters the MPLS-LDP Configuration Mode in the current context. This command configures the protocol parameters for the MPLS protocol family.
MPLS-LDP Configuration Mode	**advertise-labels { explicit-null | implicit-null }**	Configure advertisement of Implicit NULL or Explicit NULL label for all the prefixes advertised by the system in this context.
MPLS-LDP Configuration Mode	**discovery { hello { hello-interval** *seconds* **| hold-interval** *seconds* **} | transport-address** *ip_address* **}**	Configures the Label Distribution Protocol (LDP) neighbor discovery parameters.
MPLS-LDP Configuration Mode	**enable**	Enables Label Distribution Protocol (LDP).
MPLS-LDP Configuration Mode	**router-id** *ip_address*	Configures the LDP Router ID.

CLI Mode	Command	Description
MPLS-LDP Configuration Mode	**session timers { hold-interval** *seconds* **| keepalive-interval** *seconds* **}**	Configures the LDP session parameters.

Table 17: VPN-Related Monitoring Commands

CLI Mode	Command	Description
Exec Mode show Commands	**show ip bgp neighbors**	Displays information regarding BGP neighbors.
Exec Mode show Commands	**show ip bgp vpnv4 { all | route-distinguisher | vrf }**	Displays all VPNv4 routing data, routing data for a VRF or a route-distinguisher.
Exec Mode show Commands	**show ip bgp vpnv6**	Displays contents of VPNv6 routing table.
Exec Mode show Commands	**show ip bgp vpnv6 { all | route-distinguisher | vrf }**	Displays all VPNv6 routing data, routing data for a VRF or a route-distinguisher.
Exec Mode show Commands	**show ip pool**	Displays pool details including the configured VRF.
Exec Mode show Commands	**show mpls cross-connect**	Displays MPLS cross-connect information. MPLS tunnel cross-connects between interfaces and Label-Switched Paths (LSPs) connect two distant interface circuits of the same type via MPLS tunnels that use LSPs as the conduit.
Exec Mode show Commands	**show mpls ftn [vrf** *vrf_name*	Displays MPLS FEC-to-NHLFE (FTN) table information.
Exec Mode show Commands	**show mpls ftn [vrf** *vrf_name* **]**	Displays contents of the MPLS FTN table for a specified VRF.
Exec Mode show Commands	**show mpls ilm**	Displays MPLS Incoming Label Map (ILM) table information.
Exec Mode show Commands	**show mpls ldp**	Displays the MPLS LDP information.
Exec Mode show Commands	**show mpls nexthop-label-forwarding-entry**	Displays MPLS Next-Hop Label Forwarding Entry (NHLFE) table information.

Monitoring and Troubleshooting

This section provides information regarding the CLI command to monitor and troubleshoot the feature.

Show Commands and Outputs

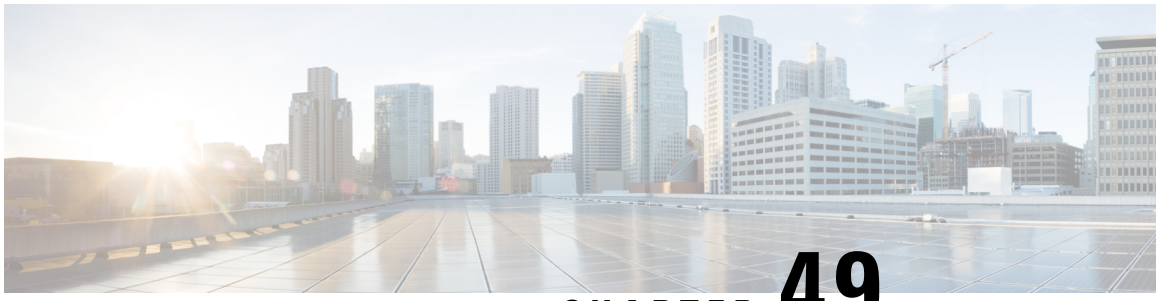
This section provides information regarding show commands and their outputs in support of this feature.

show mpls fn vpp

The output of this CLI command contains the following new field for the MPLS Support on VPC-SI for CUPS feature:

- vpp
  - all-vrf
  - summary
  - vrf

Note This new field enables viewing of the VPP dataplane values that are confiigured in the VPP dataplane forwarder. This show command is used for debugging along with the existing debug commands.



CHAPTER 49

Multiple Control Plane Support on User Plane

- Revision History, on page 389
- Feature Description, on page 389
- How it Works, on page 390
- Configuring Multiple Control Plane Support on User Plane, on page 392
- Monitoring and Troubleshooting, on page 393
- Sample RCM Configuration, on page 398

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
With this release, the feature is enhanced to support connection of upto eight CPs with a single UP.	21.26
With this release, the feature is enhanced to support connection of upto five CPs with a single UP.	21.25
First introduced.	Pre 21.24

Feature Description

In releases prior to 21.19.1, the CUPS architecture supported only a single Sx interface between User Plane (UP) and Control Plane (CP). In 21.19.1 and later releases, this feature enables single UP to establish multiple Sx interfaces to multiple CPs. Multiple Sx peers in a CP group are configured on UP to establish multiple Sx associations between a single UP and multiple CPs.

When Multiple CPs are connected to single UP, it allows a subscriber to connect to UP using any of the available CP. One of the primary use case of Multiple Sx feature is Active-Active redundancy. Even though it does not offer redundancy, as the calls are not recovered, multiple Sx allows the UPs connected to one CP

to be still accessible in case of a CP failure. If a CP fails, the calls serviced by that CP are lost. When they re-attach, the calls are routed to other available CPs which reuses the same UP pool.

In 21.20 and later releases, the feature supports configuration of same APN, and all related configuration, across multiple CPs so that the subscriber can attach using any of the available CP.

The Sx IP pool update message contains the CP address to enable UP VPNMgr to distinguish between routes installed from various CPs.

Note

- Both CP and UP are separately configured.
- Instead of a PFD push, the Redundancy and Configuration Management (RCM) pushes the configuration on UP.
- It's recommended not to configure more than four CP peer IP addresses in a single CP group.

How it Works

To configure multiple CPs with different Active Charging System (ACS) service, this feature leverages Redundancy and Configuration Management (RCM) functionality to push a super-set of configuration to UP.

Prerequisites

The following prerequisites must be met to configure multiple CPs:

- **Ruledef**:

  UP provides UE service with different rule definition (Ruledef) configurations on multiple CPs under the same ACS (ECS) service. However, the Ruledef with the same name on different CPs must be common. For example, the following table shows Ruledef configurations on multiple CPs.

CP1	CP2	CP3	CP4
Rule_def 1	Rule_def 1	Rule_def 2	Rule_def 2
Rule_def 3	Rule_def 4	Rule_def 5	Rule_def 6

- **Group-of-Ruledefs (GoR)**:

  UP provides UE service with different Group-of-Ruledefs (GoR) configurations on multiple CPs under the same ACS (ECS) service. However, the GoR with the same name on different CPs must be common. For example, the following table shows GoR configurations on multiple CPs.

CP1	CP2	CP3	CP4
GoR 1	GoR 1	GoR 2	GoR 2
GoR 3	GoR 4	GoR 5	GoR 6

- **Rulebase**:

UP provides UE service with different Rulebase (RB) configurations on multiple CPs under the same ACS (ECS) service. However, the rulebase with the same name on different CPs must be common. For example, the following table shows Rulebase configurations on multiple CPs.

CP1	CP2	CP3	CP4
RB 1	RB 1	RB 2	RB 2
RB 3	RB 4	RB 5	RB 6

- **IP Pools**:

  Each CP must be configured with mutually exclusive IP pools. This is to ensure that the unique IP address is assigned to subscriber when subscribers with same APN are serviced by different CPs. For example, the following table shows IP Pool configurations on multiple CPs.

CP1	CP2	CP3	CP4
Pool 1	Pool 2	Pool 3	Pool 4

  Each CP pushes IP Pool configuration to UP during Sx Association procedure.

UP1	UP2
Pool 1	Pool 1
Pool 2	Pool 2
Pool 3	Pool 3
Pool 4	Pool 4

- **APN**:

  UP provides UE service with different APN definition configurations on multiple CPs. However, the APN definition with the same name on different CPs, must be common and they should point to the same egress context. For example, the following table shows APN configurations on multiple CPs.

CP1	CP2	CP3	CP4
APN 1	APN 1	APN 2	APN 2
APN 3	APN 4	APN 5	APN 6

- **Egress Context**

  Each CP must be configured with same context name which is configured as egress context for APNs configured in that CP. UP must be configured with all the egress context present on different CPs, to push the IP pools from CP to specific egress context on UP. For example, the following table shows egress context configurations on multiple CPs.

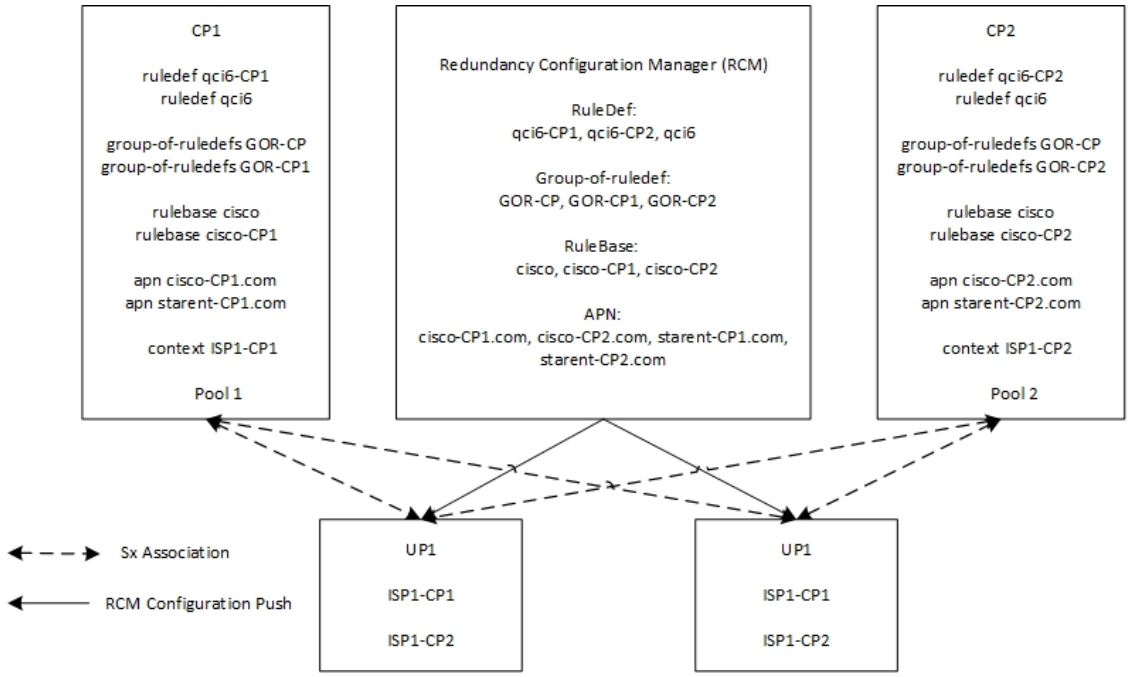
CP1	CP2	CP3	CP4
ISP1	ISP1	ISP1	ISP1

  For example, the following table shows egress context configurations on multiple UPs.

UP1	UP2
ISP1	ISP1

The following image shows a sample RCM configuration of two CPs communicating with two UPs.

Figure 20: Sample RCM Configuration of Two CPs Communicating with Two UPs



Configuring Multiple Control Plane Support on User Plane

This section provides information about CLI commands that are available in support of this feature.

Disabling PFD Configuration Push from CP

As configuration push to UP is done through RCM, use the following CLI commands to disable PFD configuration push from CP.

```
configure
  user-plane-group group_name
    sx-pfd-push disabled
    end
```

Configuring Multiple CP on UP

Use the following CLI commands to configure multiple CP on UP by adding multiple peer node under Control Plane Group Configuration mode.

```
configure
  control-plane-group group_name
    peer-node-id ipv4-address ipv4_address
    peer-node-id ipv4-address ipv4_address
    end
```

Monitoring and Troubleshooting

This section provides information about monitoring and troubleshooting the Multiple CP Support on UP feature.

Show Commands and/or Outputs

This section describes the show commands that are available in support of this feature.

show sx-service statistics address <ip_address>

Use this command to display Sx statistics for a CP node. The following is a sample output:

```
Session Management Messages:
Session Establishment Request:
 Total TX:                          0   Total RX:                          2
 Initial TX:                        0   Initial RX:                        2
 Retrans TX:                        0   Retrans RX:                        0
 Discarded:                         0
 No Rsp RX:                         0
 Throttled:                         0

Session Establishment Response:
 Total TX:                          2   Total RX:                          0
 Initial TX:                        2   Initial RX:                        0
 Accepted:                          2   Accepted:                          0
 Denied:                            0   Denied:                            0
 Retrans TX:                        0   Discarded:                         0

Session Modification Request:
 Total TX:                          0   Total RX:                         10
 Initial TX:                        0   Initial RX:                       10
 Retrans TX:                        0   Retrans RX:                        0
 Discarded:                         0   Intf Type Mismatch:                0
 No Rsp RX:                         0

Session Modification Response:
 Total TX:                         10   Total RX:                          0
 Initial TX:                       10   Initial RX:                        0
 Accepted:                         10   Accepted:                          0
 Denied:                            0   Denied:                            0
 Retrans TX:                        0   Discarded:                         0

Session Deletion Request:
 Total TX:                          0   Total RX:                          2
 Initial TX:                        0   Initial RX:                        2
 Retrans TX:                        0   Retrans RX:                        0
 Discarded:                         0
 No Rsp RX:                         0

Session Deletion Response:
 Total TX:                          2   Total RX:                          0
 Accepted:                          2   Accepted:                          0
 Denied:                            0   Denied:                            0
 Discarded:                         0

Session Report Request:
 Total TX:                          3   Total RX:                          0
 Initial TX:                        3   Initial RX:                        0
```

```
 Retrans TX:                      0   Retrans RX:                      0
 Discarded:                       0
 No Rsp RX:                       0

Session Report Response:
 Total TX:                        0   Total RX:                        3
 Initial TX:                      0   Initial RX:                      3
 Accepted:                        0   Accepted:                        3
 Denied:                          0   Denied:                          0
 Retrans TX:                      0   Discarded:                       0

Node Management Messages:
Prime PFD Management Request:
 Total TX:                        0   Total RX:                        0
 Initial TX:                      0   Initial RX:                      0
 Retrans TX:                      0   Retrans RX:                      0
 No Rsp received TX:              0   Discarded:                       0

Prime PFD Management Response:
 Total TX:                        0   Total RX:                        0
 Initial TX:                      0   Initial RX:                      0
 Accepted:                        0   Accepted:                        0
 Denied:                          0   Denied:                          0
 Retrans TX:                      0   Discarded:                       0

Association Setup Request:
 Total TX:                        1   Total RX:                        0
 Initial TX:                      1   Initial RX:                      0
 Retrans TX:                      0   Retrans RX:                      0
 No Rsp received TX:              0   Discarded:                       0

Association Setup Response:
 Total TX:                        0   Total RX:                        1
 Initial TX:                      0   Initial RX:                      1
 Accepted:                        0   Accepted:                        1
 Denied:                          0   Denied:                          0
 Retrans TX:                      0   Discarded:                       0

Association Update Request:
 Total TX:                        0   Total RX:                        3
 Initial TX:                      0   Initial RX:                      3
 Retrans TX:                      0   Retrans RX:                      0
 No Rsp received TX:              0   Discarded:                       0

Association Update Response:
 Total TX:                        3   Total RX:                        0
 Initial TX:                      3   Initial RX:                      0
 Accepted:                        3   Accepted:                        0
 Denied:                          0   Denied:                          0
 Retrans TX:                      0   Discarded:                       0

Association Release Request:
 Total TX:                        0   Total RX:                        0
 Initial TX:                      0   Initial RX:                      0
 Retrans TX:                      0   Retrans RX:                      0
 No Rsp received TX:              0   Discarded:                       0

Association Release Response:
 Total TX:                        0   Total RX:                        0
 Initial TX:                      0   Initial RX:                      0
 Accepted:                        0   Accepted:                        0
 Denied:                          0   Denied:                          0
 Retrans TX:                      0   Discarded:                       0
```

```
Node Report Request:
 Total TX:                          0   Total RX:                          0
 Initial TX:                        0   Initial RX:                        0
 Retrans TX:                        0   Retrans RX:                        0
 No Rsp received TX:                0   Discarded:                         0

Node Report Response:
 Total TX:                          0   Total RX:                          0
 Initial TX:                        0   Initial RX:                        0
 Accepted:                          0   Accepted:                          0
 Denied:                            0   Denied:                            0
 Retrans TX:                        0   Discarded:                         0

Heartbeat Request:
 Total TX:                       1398   Total RX:                       1398
 Initial TX:                     1398   Initial RX:                     1398
 Retrans TX:                        0

Heartbeat Response:
 Total TX:                       1398   Total RX:                       1398

Stats framework related messages:
Stats Query Request:
 Total TX:                          0   Total RX:                          0
 Initial TX:                        0   Initial RX:                        0
 Retrans TX:                        0   Retrans RX:                        0
 No Rsp received TX:                0   Discarded:                         0

Stats Query Response:
 Total TX:                          0   Total RX:                          0
 Initial TX:                        0   Initial RX:                        0
 Accepted:                          0   Accepted:                          0
 Denied:                            0   Denied:                            0
 Retrans TX:                        0   Discarded:                         0

Stats Query Acknowledgement:
 Total TX:                          0   Total RX:                          0
 Initial TX:                        0   Initial RX:                        0
 Retrans TX:                        0   Retrans RX:                        0
 Discarded:                         0

Total Signalling Packets:
         TX:                       21           RX:                       21

Total Signalling Bytes:
         TX:                     2092           RX:                     5381
```

Use the **clear sx-service statistics address** *ip_address* CLI command to clear Sx statistics for a CP node.

show user-plane-service statistics peer-address <ip_address>

Use this command to display the node-level service statistics for a UP. The following is a sample output:

```
Peer IP                    : 209.165.200.225

Subscribers Total:
PDNs Total:
Active:                         1        Setup:                         1
Released:                       0        Rejected:                      0

PDNs By PDN-Type:
IPv4 PDNs:
Active:                         1        Setup:                         1
Released:                       0
```

```
IPv6 PDNs:
Active:                          0        Setup:                          0
Released:                        0

IPv4v6 PDNs:
Active:                          0        Setup:                          0
Released:                        0

eMPS PDNs Total:
Active:                          0        Setup:                          0
Released:                        0        Rejected:                       1

PDNs By interface-Type:
Sxa interface-type PDNs:
Active:                          0        Setup:                          0
Released:                        0

Sxb interface-type PDNs:
Active:                          1        Setup:                          1
Released:                        0

Sxab interface-type PDNs:
Active:                          0        Setup:                          0
Released:                        0

N4 interface-type PDNs:
Active:                          0        Setup:                          0
Released:                        0

PDNs Rejected By Reason:
No Resource:                     0        Missing or unknown APN:         0
Addr not alloc:                  0        Addr not present:               0
No memory available:             0        System Failure:                 0
Rule install failed:             0        SFW policy mismatch:            0

PDNs Released By Reason:
Network initiated release:       0        Admin disconnect:               0

Total Data Statistics:
Uplink :                                  Downlink :
Total Pkts:                      0        Total Pkts:                     0
Total Bytes:                     0        Total Bytes:                    0
Total Dropped Pkts:              0        Total Dropped Pkts:             0
Total Dropped Bytes:             0        Total Dropped Bytes:            0

Data Statistics Per PDN-Type:
IPv4 PDNs:
Uplink :                                  Downlink :
Total Pkts:                      0        Total Pkts:                     0
Total Bytes:                     0        Total Bytes:                    0

IPv6 PDN Data Statistics:
Uplink :                                  Downlink :
Total Pkts:                      0        Total Pkts:                     0
Total Bytes:                     0        Total Bytes:                    0

IPv4v6 PDN Data Statistics:
Uplink :                                  Downlink :
Total Pkts v4:                   0        Total Pkts v4:                  0
Total Bytes v4:                  0        Total Bytes v4:                 0
Total Pkts v6:                   0        Total Pkts v6:                  0
Total Bytes v6:                  0        Total Bytes v6:                 0
```

Use the **clear user-plane-service statistics peer-address** *ip_address* CLI command to clear the node-level service statistics for a UP.

show ip chunks peer <ip_address>

Use this command to display per CP IPv4 pool chunks at UP. The following is a sample output.

```
==============================================================================================
Peer Address: 1.0.0.1
==============================================================================================
|----------|-----------|------------------------------|--------------|--------------|-----------|
| chunk-id | chunk-size |              vrf-name          |   start-addr  |    end-addr   |
 used-addrs |
|----------|-----------|------------------------------|--------------|--------------|-----------|
|   1048576|        1024|                              |      192.0.2.1|      192.0.2.2|
           0|
|   1048577|        1024|                              |      192.0.2.3|      192.0.2.4|
           0|
|   1048578|        1024|                              |      192.0.2.5|      192.0.2.6|
           0|
|   3145728|        1024|                         vrf1|      192.0.2.7|      192.0.2.8|
           0|
|   3145729|        1024|                         vrf1|      192.0.2.9|     192.0.2.10|
           0|
|   3145730|        1024|                         vrf1|     192.0.2.11|     192.0.2.12|
           0|
|   4194304|        1024|                             |     192.0.2.13|     192.0.2.14|
           0|
|   4194305|        1024|                             |     192.0.2.15|     192.0.2.16|
           0|
|   4194306|        1024|                             |     192.0.2.17|     192.0.2.18|
           0|
|----------|-----------|------------------------------|--------------|--------------|-----------|
```

show ipv6 chunks peer <ip_address>

Use this command to display per CP IPv6 pool chunks at UP. The following is a sample output.

```
==============================================================================================
Peer Address: 1.0.0.101
==============================================================================================
|----------|-----------|-----------------|--------------------|--------------------|--------------|
| chunk-id | chunk-size |     vrf-name     | start-prefix       |   end-prefix       |
used-prefixes |
|----------|-----------|-----------------|--------------------|--------------------|--------------|
|2098200576|        1024|                 |              3001::|      3001:0:0:3ff::|
           0|
|2098200577|        1024|                 |      3001:0:0:400::|      3001:0:0:7ff::|
           0|
|2098200578|        1024|                 |      3001:0:0:800::|      3001:0:0:bff::|
           0|
|2099249152|        1024|             vrf1|              4001::|      4001:0:0:3ff::|
           0|
|2099249153|        1024|             vrf1|      4001:0:0:400::|      4001:0:0:7ff::|
           0|
|2099249154|        1024|             vrf1|      4001:0:0:800::|      4001:0:0:bff::|
           0|
|----------|-----------|-----------------|--------------------|--------------------|--------------|
```

Sample RCM Configuration

The following is a sample RCM configuration to configure the feature.

```
configure
etcd replicas 1
endpoint rcm-chkptmgr
 replicas 7
 vip-ip 209.165.200.225
exit
endpoint rcm-configmgr
 vip-ip 209.165.200.225
exit
endpoint rcm-bfdmgr
 vip-ip 209.165.200.226
exit
endpoint rcm-controller
 vip-ip 209.165.200.225
exit
logging level application trace
logging level transaction trace
logging level tracing off
logging name infra.config.core level application trace
logging name infra.config.core level transaction trace
logging name infra.resource_monitor.core level application debug
logging name infra.resource_monitor.core level transaction debug
k8 smf profile rcm-config-ep disable-cm apn gtpp creditCtrl packetFilter urrList ruledef
rulebase miscacs global chargingAction upfCpg upSvcs sxService gtpuService upfIfc
lawfulIntercept apnprofile
k8 smf profile rcm-bfd-ep bfd-monitor group 1
 endpoint ipv4 209.165.200.227
 endpoint ipv4 209.165.200.228
 endpoint ipv4 209.165.200.229
 standby 1
exit
system mode running
helm default-repository smf
helm repository smf
 access-token
dev-deployer.gen:AKCp5ekcXA7TknM9DbLASNBw4jwVEsx9Z9WpQwEvCvCQ2mJhLymcz6BfbH38YJiWC6fn1cKmw
 url http://example.com
exit
k8s name          unknown
k8s namespace     rcm
k8s nf-name       rcm
k8s registry      dockerhub.xxx.com/smi-fuse-docker-internal
k8s single-node   false
k8s use-volume-claims false
k8s ingress-host-name 209.165.200.225.nip.io
profile smf rcm
 node-id 123456
exit
svc-type upinterface
svc-type sxsvc
svc-type upsvc
svc-type gtpusvc
svc-type cpgrp
 redundancy-group 1
  host 209.165.200.225:22
host 295 "config "
host 296 "control-plane-group CPGROUP21 "
host 297 "peer-node-id ipv4-address 209.165.200.230 "
```

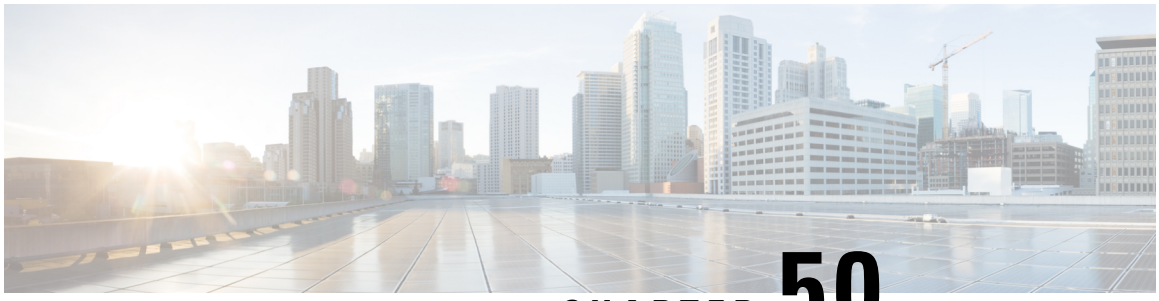
```
host 298 "peer-node-id ipv4-address 209.165.200.231 "
host 299 "exit "
host 300 "end "
exit
exit
svc-type sxsvc
svc-type upsvc
svc-type gtpusvc
svc-type cpgrp
 redundancy-group 1
  host 209.165.200.225:22
host 393 " config "
host 394 "control-plane-group CPGROUP21 "
host 395 "peer-node-id ipv4-address 209.165.200.230 "
host 396 "peer-node-id ipv4-address 209.165.200.231 "
host 397 "48 exit "
host 398 "49 end "
exit
exit
exit
redundancy-group 1
common 1 " sleep 5  "
common 2 " config "
common 3 "   active-charging service ACS "
common 4 "     #exit "
common 5 "     ruledef ipv6 "
common 6 "       icmpv6 any-match = TRUE "
common 7 "     #exit "
common 8 "     ruledef qci1 "
common 9 "       tcp src-port = 1001 "
common 10 "     #exit "
common 11 "     ruledef qci2 "
common 12 "       tcp src-port = 1002 "
common 13 "     #exit "
common 14 "     ruledef qci6 "
common 15 "       tcp src-port = 1006 "
common 16 "     #exit "
common 17 "     ruledef qci6-CP1 "
common 18 "       udp src-port = 1010 "
common 19 "     #exit "
common 20 "     ruledef qci6-CP2 "
common 21 "       udp src-port = 1020 "
common 22 "     #exit "
common 23 "     group-of-ruledefs GOR "
common 24 "       add-ruledef priority 11 ruledef qci1 "
common 25 "       add-ruledef priority 22 ruledef qci2 "
common 26 "       add-ruledef priority 33 ruledef ipv6 "
common 27 "     #exit "
common 28 "     group-of-ruledefs GOR-CP1 "
common 29 "       add-ruledef priority 11 ruledef qci1 "
common 30 "       add-ruledef priority 33 ruledef ipv6 "
common 31 "     #exit "
common 32 "     group-of-ruledefs GOR-CP2 "
common 33 "       add-ruledef priority 11 ruledef qci2 "
common 34 "       add-ruledef priority 33 ruledef ipv6 "
common 35 "     #exit "
common 36 "     packet-filter ipv6 "
common 37 "       ip protocol = 58 "
common 38 "       priority 22 "
common 39 "     #exit "
common 40 "     packet-filter qci1 "
common 41 "       ip protocol = 6 "
common 42 "       ip remote-port = 1001 "
common 43 "       priority 1 "
```

```
common 44 "     #exit "
common 45 "     packet-filter qci2 "
common 46 "       ip protocol = 6 "
common 47 "       ip remote-port = 1002 "
common 48 "       priority 2 "
common 49 "     #exit "
common 50 "     packet-filter qci6 "
common 51 "       ip protocol = 6 "
common 52 "       ip remote-port = 1006 "
common 53 "       priority 6 "
common 54 "     #exit "
common 55 "     packet-filter qci6-CP1 "
common 56 "       ip protocol = 17 "
common 57 "       ip remote-port = 1010 "
common 58 "       priority 1 "
common 59 "     #exit "
common 60 "     packet-filter qci6-CP2 "
common 61 "       ip protocol = 17 "
common 62 "       ip remote-port = 1020 "
common 63 "       priority 1 "
common 64 "     #exit "
common 65 "     urr-list URR_ID_LIST "
common 66 "       rating-group 1 urr-id 1 "
common 67 "       rating-group 2 urr-id 2 "
common 68 "       rating-group 3 urr-id 3 "
common 69 "       rating-group 4 urr-id 4 "
common 70 "       rating-group 5 urr-id 5 "
common 71 "       rating-group 6 urr-id 6 "
common 72 "       rating-group 7 urr-id 7 "
common 73 "       rating-group 8 urr-id 8 "
common 74 "       rating-group 9 urr-id 9 "
common 75 "       rating-group 10 urr-id 10 "
common 76 "       rating-group 11 urr-id 11 "
common 77 "       rating-group 12 urr-id 12 "
common 78 "       rating-group 13 urr-id 13 "
common 79 "       rating-group 14 urr-id 14 "
common 80 "     #exit "
common 81 "     charging-action ipv6 "
common 82 "       content-id 11 "
common 83 "       billing-action egcdr "
common 84 "       billing-action rf "
common 85 "       cca charging credit rating-group 11 "
common 86 "       qos-class-identifier 5 "
common 87 "       flow limit-for-bandwidth id 2 "
common 88 "       tft packet-filter ipv6 "
common 89 "     #exit "
common 90 "     charging-action qci1 "
common 91 "       content-id 1 "
common 92 "       billing-action egcdr "
common 93 "       billing-action rf "
common 94 "       cca charging credit rating-group 1 "
common 95 "       qos-class-identifier 1 "
common 96 "       flow limit-for-bandwidth id 1 "
common 97 "       allocation-retention-priority 1 pvi 0 pci 1 "
common 98 "       tft packet-filter qci1 "
common 99 "     #exit "
common 100 "     charging-action qci1-GOR "
common 101 "       content-id 1 "
common 102 "       billing-action egcdr "
common 103 "       billing-action rf "
common 104 "       cca charging credit rating-group 1 "
common 105 "       qos-class-identifier 1 "
common 106 "       flow limit-for-bandwidth id 1 "
common 107 "       allocation-retention-priority 1 pvi 0 pci 1 "
```

```
common 108 "       tft packet-filter ipv6 "
common 109 "       tft packet-filter qci1 "
common 110 "       tft packet-filter qci2 "
common 111 "     #exit "
common 112 "     charging-action qci1-GOR-CP1 "
common 113 "       content-id 1 "
common 114 "       billing-action egcdr "
common 115 "       billing-action rf "
common 116 "       cca charging credit rating-group 1 "
common 117 "       qos-class-identifier 1 "
common 118 "       flow limit-for-bandwidth id 1 "
common 119 "       allocation-retention-priority 1 pvi 0 pci 1 "
common 120 "       tft packet-filter ipv6 "
common 121 "       tft packet-filter qci1 "
common 122 "     #exit "
common 123 "     charging-action qci1-GOR-CP2 "
common 124 "       content-id 1 "
common 125 "       billing-action egcdr "
common 126 "       billing-action rf "
common 127 "       cca charging credit rating-group 1 "
common 128 "       qos-class-identifier 1 "
common 129 "       flow limit-for-bandwidth id 1 "
common 130 "       allocation-retention-priority 1 pvi 0 pci 1 "
common 131 "       tft packet-filter ipv6 "
common 132 "       tft packet-filter qci2 "
common 133 "     #exit "
common 134 "     charging-action qci2 "
common 135 "       content-id 2 "
common 136 "       billing-action egcdr "
common 137 "       billing-action rf "
common 138 "       cca charging credit rating-group 2 "
common 139 "       qos-class-identifier 2 "
common 140 "       flow limit-for-bandwidth id 1 "
common 141 "       allocation-retention-priority 2 pvi 0 pci 1 "
common 142 "       tft packet-filter qci2 "
common 143 "     #exit "
common 144 "     charging-action qci6 "
common 145 "       content-id 6 "
common 146 "       billing-action egcdr "
common 147 "       billing-action rf "
common 148 "       cca charging credit rating-group 6 "
common 149 "       qos-class-identifier 6 "
common 150 "       flow limit-for-bandwidth id 2 "
common 151 "       allocation-retention-priority 6 pvi 0 pci 1 "
common 152 "       tft packet-filter qci6 "
common 153 "     #exit "
common 154 "     charging-action qci6-CP1 "
common 155 "       content-id 12 "
common 156 "       billing-action egcdr "
common 157 "       billing-action rf "
common 158 "       cca charging credit rating-group 12 "
common 159 "       qos-class-identifier 6 "
common 160 "       flow limit-for-bandwidth id 2 "
common 161 "       allocation-retention-priority 6 pvi 0 pci 1 "
common 162 "       tft packet-filter qci6-CP1 "
common 163 "     #exit "
common 164 "     charging-action qci6-CP2 "
common 165 "       content-id 13 "
common 166 "       billing-action egcdr "
common 167 "       billing-action rf "
common 168 "       cca charging credit rating-group 13 "
common 169 "       qos-class-identifier 6 "
common 170 "       flow limit-for-bandwidth id 2 "
common 171 "       allocation-retention-priority 6 pvi 0 pci 1 "
```

```
common 172 "       tft packet-filter qci6-CP2 "
common 173 "     #exit "
common 174 "     bandwidth-policy bw_policy1 "
common 175 "       flow limit-for-bandwidth id 1 group-id 1 "
common 176 "       flow limit-for-bandwidth id 2 group-id 2 "
common 177 "       group-id 1 direction downlink peak-data-rate 256000 peak-burst-size
768000 violate-action discard committed-data-rate 128000 committed-burst-size 384000 "
common 178 "       group-id 1 direction uplink peak-data-rate 256000 peak-burst-size 768000
 violate-action discard committed-data-rate 128000 committed-burst-size 384000 "
common 179 "       group-id 2 direction downlink peak-data-rate 256000 peak-burst-size
768000 violate-action discard "
common 180 "       group-id 2 direction uplink peak-data-rate 256000 peak-burst-size 768000
 violate-action discard "
common 181 "     #exit "
common 182 "     rulebase cisco "
common 183 "       billing-records egcdr "
common 184 "       action priority 1 dynamic-only group-of-ruledefs GOR charging-action
qci1-GOR "
common 185 "       action priority 11 dynamic-only ruledef qci1 charging-action qci1 "
common 186 "       action priority 22 dynamic-only ruledef qci2 charging-action qci2 "
common 187 "       action priority 66 dynamic-only ruledef qci6 charging-action qci6 "
common 188 "       action priority 666 dynamic-only ruledef ipv6 charging-action ipv6 "
common 189 "       egcdr threshold interval 3600 "
common 190 "       egcdr threshold volume total 100000 "
common 191 "       bandwidth default-policy bw_policy1 "
common 192 "     #exit "
common 193 "     rulebase cisco-CP1 "
common 194 "       billing-records egcdr "
common 195 "       action priority 1 dynamic-only group-of-ruledefs GOR-CP1 charging-action
 qci1-GOR-CP1 "
common 196 "       action priority 11 dynamic-only ruledef qci1 charging-action qci1 "
common 197 "       action priority 22 dynamic-only ruledef qci2 charging-action qci2 "
common 198 "       action priority 66 dynamic-only ruledef qci6-CP1 charging-action qci6-CP1
 "
common 199 "       action priority 666 dynamic-only ruledef ipv6 charging-action ipv6 "
common 200 "       egcdr threshold interval 1000 "
common 201 "       egcdr threshold volume total 100000 "
common 202 "       bandwidth default-policy bw_policy1 "
common 203 "     #exit "
common 204 "     rulebase cisco-CP2 "
common 205 "       billing-records egcdr "
common 206 "       action priority 1 dynamic-only group-of-ruledefs GOR-CP2 charging-action
 qci1-GOR-CP2 "
common 207 "       action priority 11 dynamic-only ruledef qci1 charging-action qci1 "
common 208 "       action priority 22 dynamic-only ruledef qci2 charging-action qci2 "
common 209 "       action priority 66 dynamic-only ruledef qci6-CP2 charging-action qci6-CP2
 "
common 210 "       action priority 666 dynamic-only ruledef ipv6 charging-action ipv6 "
common 211 "       egcdr threshold interval 1000 "
common 212 "       egcdr threshold volume total 100000 "
common 213 "       bandwidth default-policy bw_policy1 "
common 214 "     #exit "
common 215 "     rulebase default "
common 216 "     #exit "
common 217 "     credit-control group default "
common 218 "       diameter origin endpoint PGW-Gy "
common 219 "       diameter peer-select peer PGW-Gy-server "
common 220 "       quota time-threshold 10 "
common 221 "       diameter pending-timeout 150 deciseconds msg-type any "
common 222 "       diameter session failover "
common 223 "       trigger type rat qos sgsn serving-node "
common 224 "       pending-traffic-treatment noquota pass "
common 225 "       pending-traffic-treatment quota-exhausted buffer "
common 226 "       timestamp-rounding floor "
```

```
common 227 "     #exit "
common 228 "     traffic-optimization-policy default "
common 229 "     #exit "
common 230 "   #exit "
common 231 " end "
common 232 " config  "
common 233 " context ISP1-CP1  "
common 234 "    apn xxx-CP1.com  "
common 235 "      ip context-name ISP1-CP1  "
common 236 "    exit  "
common 237 "    apn yyy-CP1.com  "
common 238 "      ip context-name ISP1-CP1  "
common 239 "    exit  "
common 240 " end  "
common 241 " config  "
common 242 " context ISP1-CP2  "
common 243 "    apn xxx-CP2.com  "
common 244 "      ip context-name ISP1-CP2  "
common 245 "    exit  "
common 246 "    apn yyy-CP2.com  "
common 247 "      ip context-name ISP1-CP2  "
common 248 "    exit  "
common 249 " end  "
  exit
```



CHAPTER 50

Multiple UP Group Support

- Revision History, on page 405
- Feature Description, on page 405
- How It Works, on page 407
- Configure the Multiple UP Group Support Feature, on page 407

Revision History

Table 18: Revision History

Revision Details	Release
First Introduced	21.25

Feature Description

Remote CUPS allows a progressive configuration rollout on an operator network. You can deploy and activate a pilot or canary version N+1 on a given CP or UPs pool, while the version N configuration is still active on the other CP or UP pool until the operator decides to roll out this N+1 configuration to all the CP or UP pools after the monitoring period.

Use cases for this feature are as follows:

- ECS or ADC configuration update rollout: The ability to test the configuration using one CP or UP pilot, while the other CP or UP uses the old configuration.
- New APN configuration update: Ability to test new APN configuration using a set of CP or UP pilot, while another component uses the old configuration.
- Add or remove the IP pool configuration update.

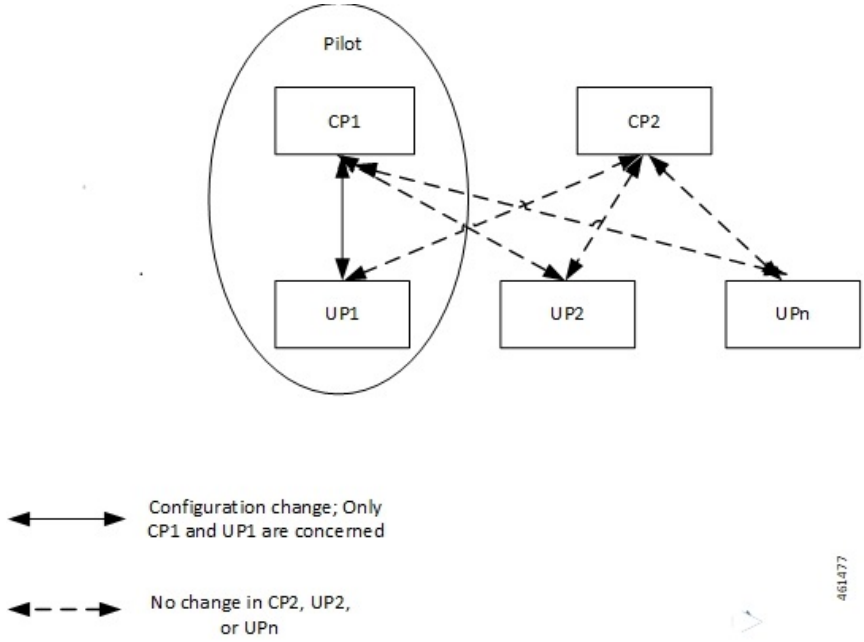
Relationships

TAC RAC profile support feature is related to the Multiple UP Group Support feature, which is used to select a test virtual APN.

Architecture

The following diagram depicts the progressive configuration rollout architecture.

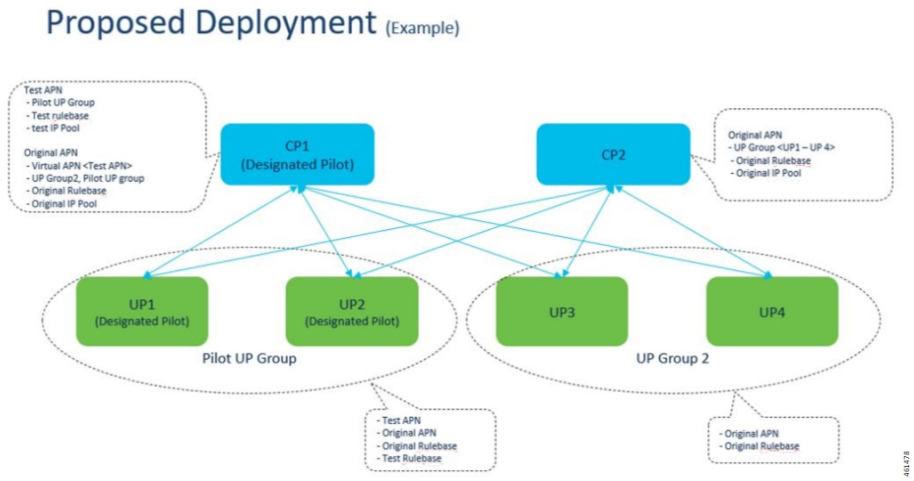
Figure 21: Progressive Configuration Rollout



Components

The following diagram depicts the proposed deployment components.

Figure 22: Proposed Deployment



How It Works

Pilot CP routes the incoming test pilot calls to the pilot UP group.

Pilot CP also routes usual business calls to any UP as per its original deployment. Therefore, this feature supports multiple UP groups under an APN. The first UP group includes the pilot UPs, and the second UP group includes all the other nonpilot UPs. A single UP cannot exist in two UP groups simultaneously. There is a strict 1:1 mapping between UP and the UP group.

Limitations and Restrictions

The Multiple UP Group Support feature has the following limitations and restrictions:

- You can apply the pilot configuration only at the UP-group level.
- Configure CP and UP independently.
- ECS configuration changes that occur at lower levels like ruledef, Charging Action, and so on, cannot be isolated from the pilot UP. Rule base level differentiation is required.
- Pilot CP and UPs must be designated at deployment. Post deployment designation requires the existing sessions to be cleared.
- If any user group is not attached to any of the APNs, then the corresponding UP nodes must be de-registered and removed from the CP configuration.
- Static-IP-Pools are not supported.

Configure the Multiple UP Group Support Feature

This section describes how to configure the Multiple UP Group Support feature.

Configuring the Multiple UP Group Support feature involves the following steps:

Serial Number	Configuration	For Pilot Configuration
1	ECS or ADC configuration (For example, ruledef, rulebase, charging action)	On Pilot CP: Configuration differentiation must be performed at the rulebase level. Changes in configuration entities like ruledef, charging action, and so on, needs duplication. On Pilot UP: Corresponding configuration changes must be performed on one or more pilot UPs directly.
2	APN configuration	• Create a new APN with desired configuration changes. • Configure the test APN as a virtual APN in the existing APN with required redirecting rules. Alternatively, use MME to redirect calls to the test APN.

Serial Number	Configuration	For Pilot Configuration
3	UP group configuration	• Select pilot CP and UPs at the time of deployment. • Enable multiple UP-groups that must be configured for an APN.
4	IP pool configuration	New IP pool: • Create a new IP Pool and associate it with the test APN. Update existing IP pool: • Perform the configuration changes to the IP pool directly. **Note:** Change cannot be localized to one or more pilot UPs only.

Use the following CLI commands to configure the UP-management policy:

```
configure
   up-mgmt-policy policy_name
      user-plane-group group_name
      end
```

Use the following configuration for UP selection on Pure-P and Collapsed call types:

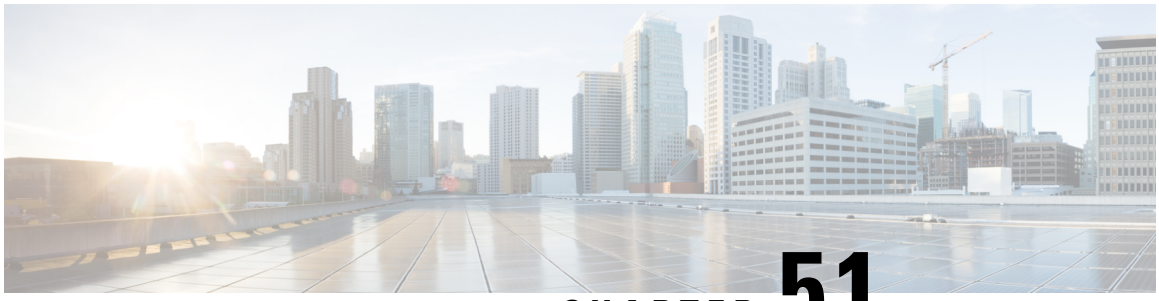
```
configure
   context context_name
      apn apn_name
         up-mgmt-policy policy_name
         end
```

Use the following configuration for UP selection on a Pure-S call type:

```
configure
   context context_name
      apn-profile  profile_name
         up-mgmt-policy policy_name
         end
```

Notes:

- *policy_name* must be a string of size 1-31.
- You can configure either the APN profile level UP-group or the UP-management policy.
- For an APN profile, you can configure only a single UP-management policy.
- Assign a pool name for the IP address allocation.



CHAPTER 51

MOCN Special Handling of CRA and CNR

- Revision History, on page 409
- Feature Description, on page 409
- TAI Change Event Handling, on page 410
- How It Works, on page 411

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

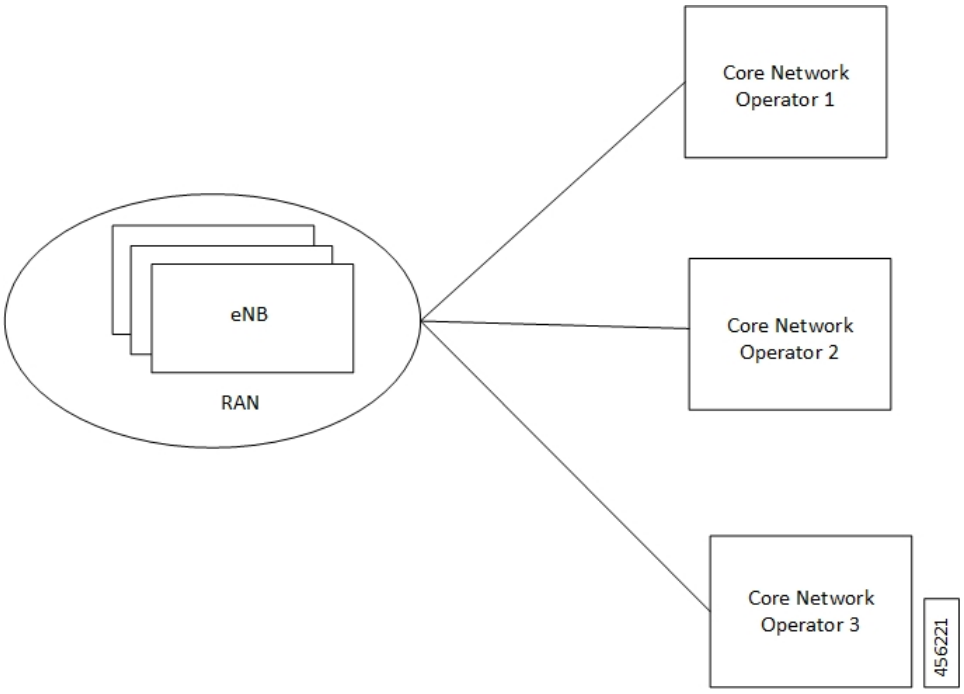
Feature Description

This feature describes the SAE-GW support to enable/disable the Multi Operator Core Network (MOCN). The feature also explains about the handling of Tracking Areas Identity (TAI) change event as requested by PCRF when MOCN is enabled.

The SAE-GW indicates the MME to Start/Stop reporting TAI change event as requested by PCRF. On receiving a TAI change from MME, the SAE-GW reports:

- The TAI change event to PCRF
- The location update to OCS

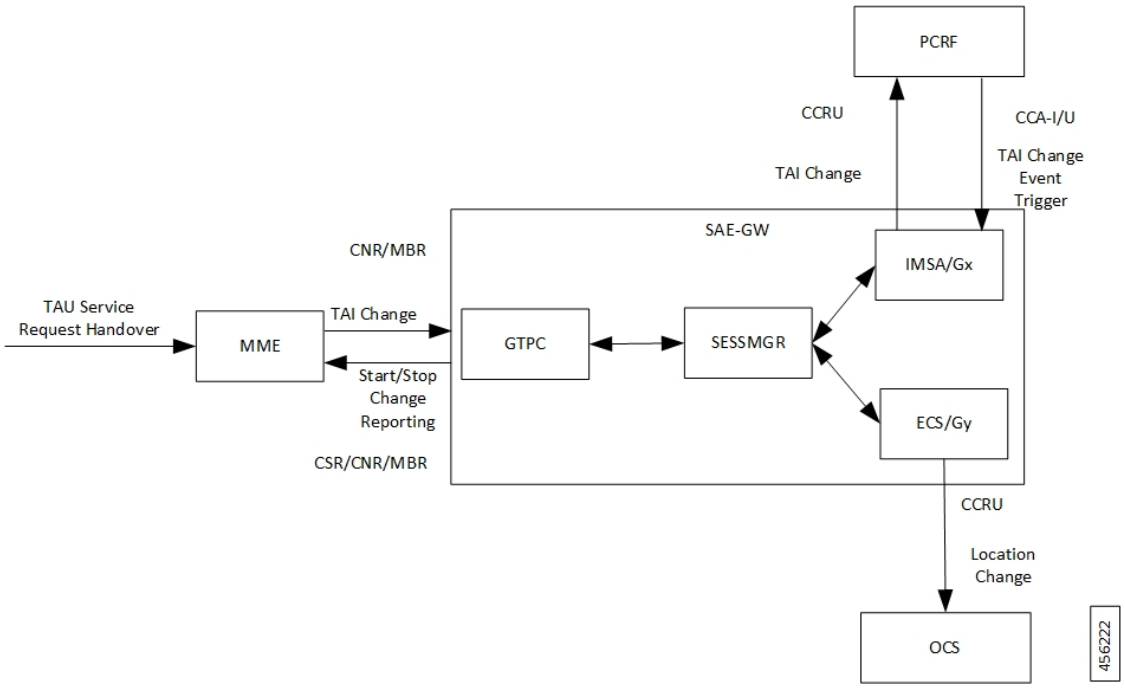
The following figure describes about the MOCN feature that allows different core network operators to connect to a shared radio access network.



TAI Change Event Handling

The following figure describes the high-level overview of the architecture of TAI change event handling.

Figure 23: TAI Change Event Handling - Process Flow



When you enable the MOCN feature on SAE-GW and it receives a TAI change trigger from PCRF in event trigger AVP of Credit Control Answer-Initial/Update (CCA-I/U), the SAE-GW sends a start reporting TAI indication to MME in Change Reporting Action (CRA) of Create Session Response/Modify Bearer Request/Change Notification Response.

The MME sends the TAI change in User Location Information of Change Notification Request/Modify Bearer Request to SAE-GW in the event of TAI change during various procedures like Tracking Area Update (TAU), Service Request and S1AP/X2 handover.

In turn, the SAE-GW indicates the TAI change to the PCRF in Event Trigger AVP and the value in 3GPP-User-Location-Info AVP of Credit Control Request-Update (CCR-U) to receive any location-dependent policies.

The SAE-GW also indicates the Location change to OCS in Trigger Type AVP and User Location Info in PS-Information AVP and Multiple Services Credit Control (MSCC) to enable location-dependent charging related procedures.

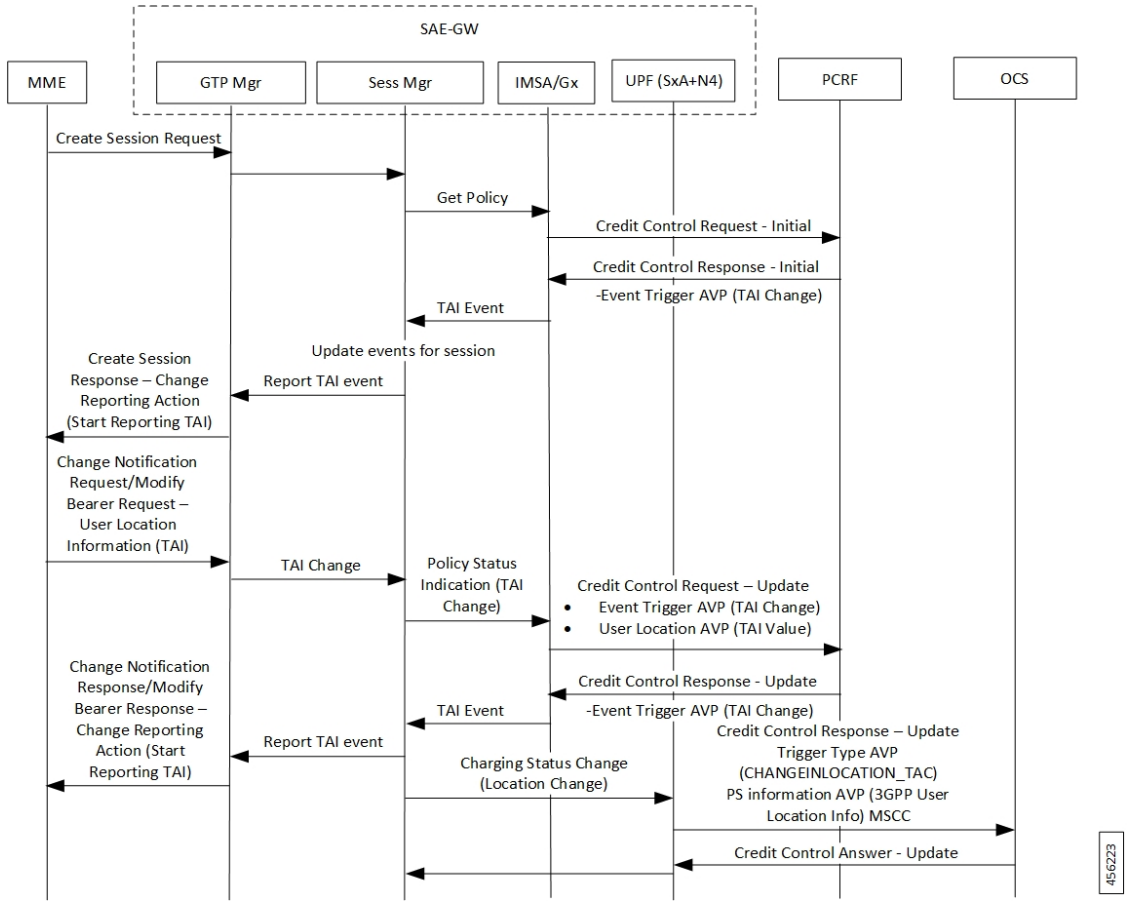
When you enable the MOCN feature on SAE-GW and it receives a No Event Trigger from PCRF in event trigger AVP of Credit Control Answer-Initial/Update (CCA-I/U), the SAE-GW sends a Stop Reporting TAI indication to MME in Change Reporting Action (CRA) of Create Session Response/Modify Bearer Request/Change Notification Response.

How It Works

The following call flows describe about the starting and stopping of the TAI change report.

Start Reporting TAI Change

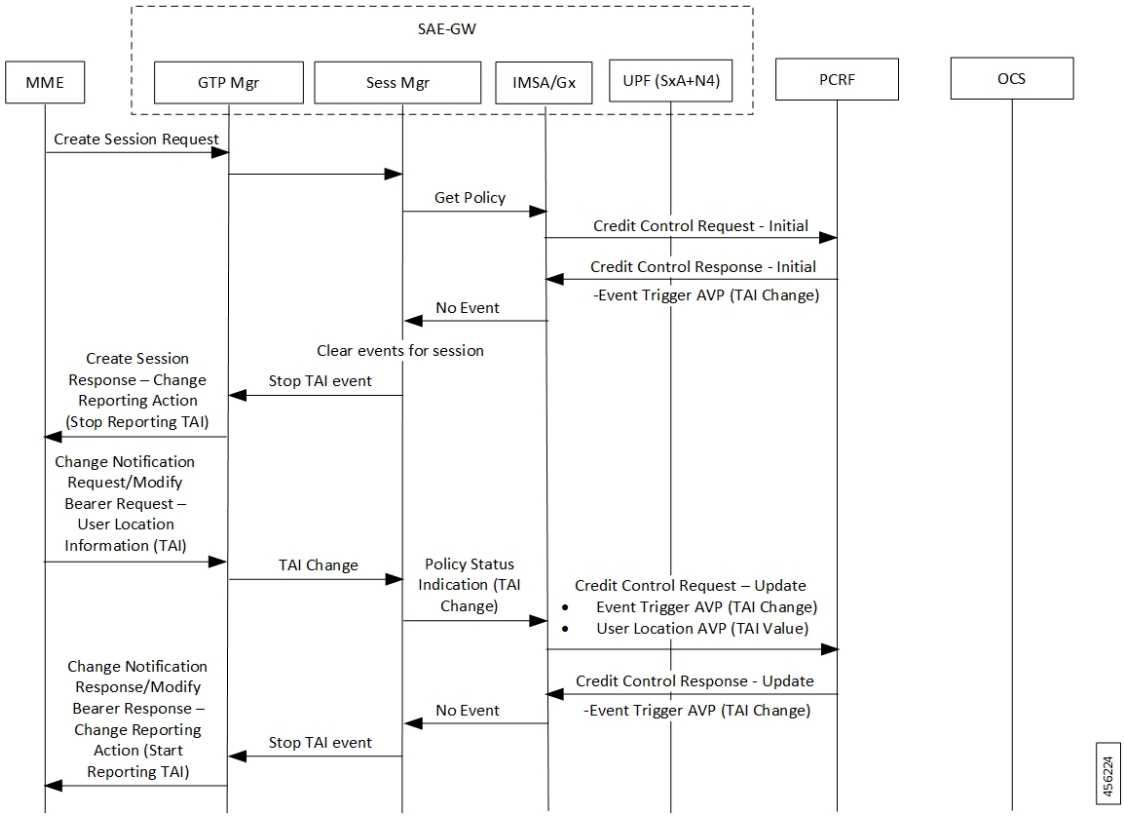
The following call flow describes about the reporting of the TAI change.



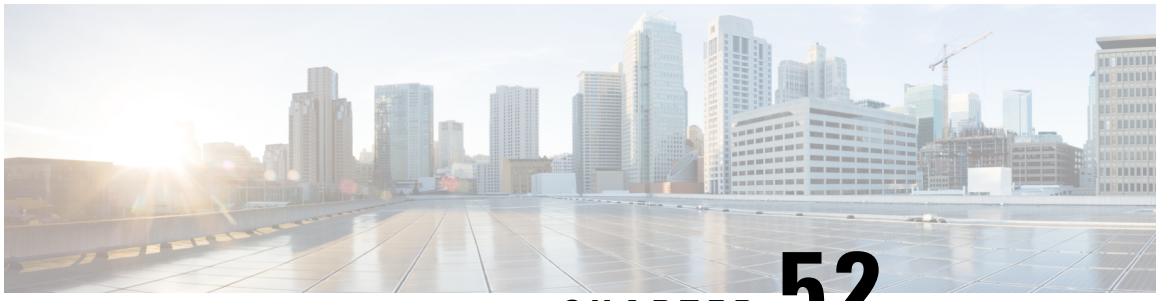
Steps	Description
1.	During the session establishment procedure, the SAE-GW sends the Create Session Response to MME with Change Reporting Action (CRA) value set to Start Reporting TAI if the PCRF requests for TAI change reporting by specifying Event Trigger set to TAI_CHANGE (26) in CCA-I to SAE-GW.
2.	The MME on detecting change in UE's TAI sends Change Notification Request or Modify Bearer Request with ULI including new TAI. The SAE-GW includes the Event-Trigger set as TAI_CHANGE (26) in CCR-U sent to PCRF and the value in User Location AVP. If the SAE-GW receives CCA-U with Event-Trigger set to TAI_CHANGE (26) from PCRF, the SAE-GW sends the Change Notification Response or Modify Bearer Response with CRA value set to Start Reporting TAI.
3.	Then the SAE-GW includes the Trigger-Type AVP with CHANGEINLOCATION_TAC (35), PS-Information AVP (3GPP-User-Location: new TAI) and MSCC in CCR-U sent to OCS and receives CCA-U from OCS.

Stop Reporting TAI Change

The following call flow describes about the reporting of the TAI change.



During the session establishment procedure, the SAE-GW sends the Create Session Response to MME with Change Reporting Action (CRA) value set to Stop Reporting TAI if the PCRF requests for no event reporting by specifying Event Trigger set to NO_EVENT_TRIGGERS in CCA-I to SAE-GW.



CHAPTER 52

N+2 UP Recovery

- Revision History, on page 415
- Feature Description, on page 415
- How It Works, on page 417
- Configuring N+2 UP Recovery, on page 432
- Monitoring and Troubleshooting, on page 434

Revision History

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In accordance with 3GPP, the CP uses Sx-based failure detection which relies on Sx keep alive message responses from the UP.

Using this approach, when the CP does not receive responses from the UP, it retransmits the Sx message a configurable number of times before declaring the UP as down and initiating session tear downs. Depending on the number of retries and the retry interval, the failure detection period can take more than 10 seconds for a reliable determination that the UP is down. Until the Sx-path failure is detected at CP, the CP continues to select the failed-UP and place new PDN-connections from UEs on the failed-UP.

In order to reduce the time it takes for the CP to detect that a UP is down, Cisco CPs can be configured to use the Bidirectional Forwarding Detection (BFD) protocol (RFC 5883 - Bidirectional Forwarding Protocol Detection (BFD) for Multihop Paths).

BFD uses significantly smaller retry periods (in the order of 200 msec) allowing for more rapid UP down detection. It is in addition to the Sx keepalive mechanism for alternate deployment scenarios (e.g. 1:1 UP redundancy).

NOTE: This feature is not dependent on Packet Flow Description (PFD) since PFD pushes common Day-N configurations across the UPs.

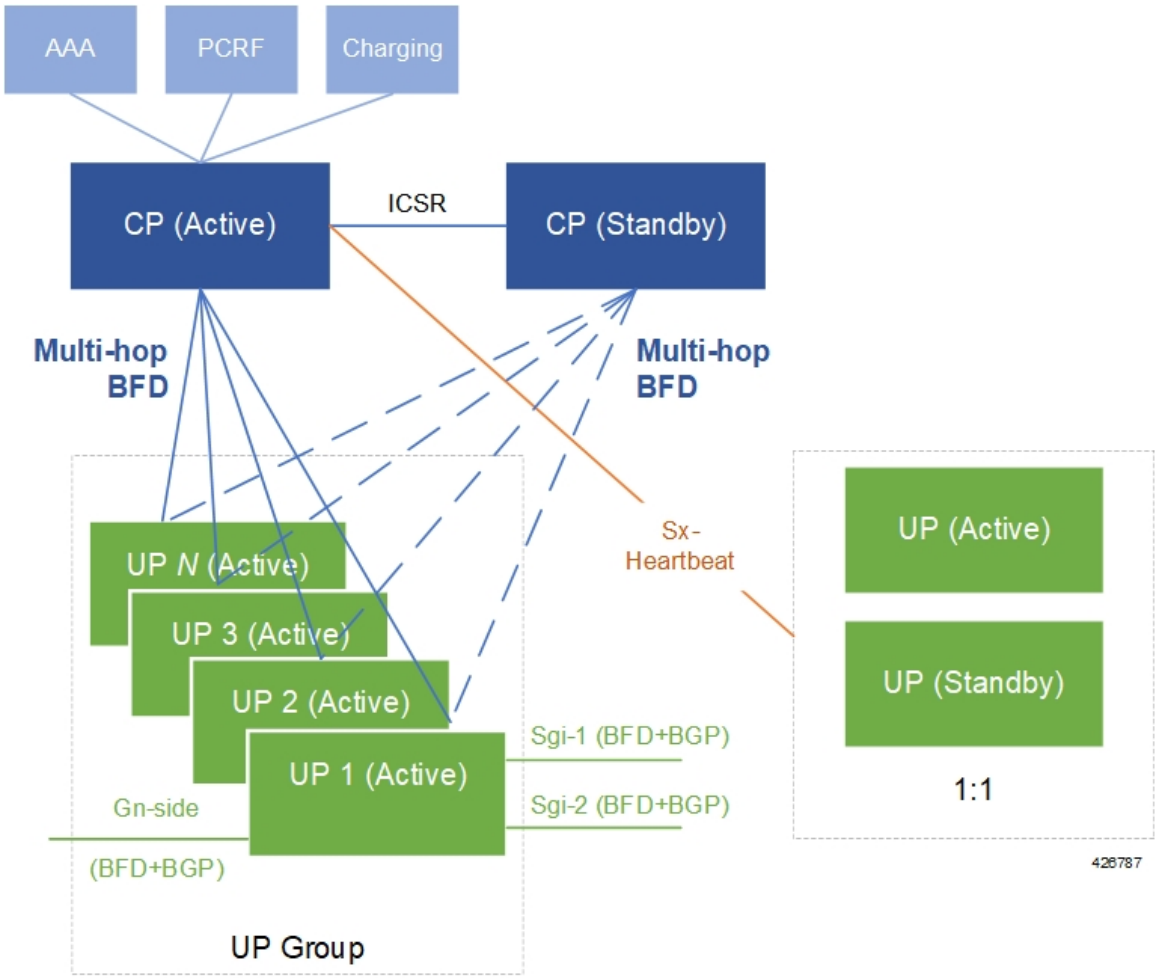
Deployment Architecture

This functionality can be enabled only in an "N+2" deployment scenario for UPs that process data sessions. In this scenario, CPs are deployed as an active-standby pair. "N" number of active UPs can be deployed to communicate with the CP. All of these UPs must be part of a specific, non-default, UP group.

NOTE: In N+2, all UPs are active. As such, this functionality only serves to improve data UP recovery times, it is not a redundancy model. It is highly recommended that UPs processing IMS traffic only be deployed in a 1:1 redundancy model.

BFD communications between the CP and UP requires the configuration of one additional loopback IP address per CP/per UP.

Figure 24: BFD Monitoring in N+2 Deployment



Limitations

- BFD-based CP failure detection is not supported in this release. CP failures can continue to be detected using the existing mechanism of Sx-path failure detection at the UP

  NOTE: It is recommended that Sx-path failure timers be configured more aggressively to more quickly prevent stale UP sessions.

- BGP monitoring on Gi/Gn interface (of UP) is not supported.
- Multi-BFD is not supported.
- BFD must be configured in the same context in which Sx is configured (Gn-side) on both the CP and UP.

How It Works

The figure and the table that follow provide a high-level description of the session detach and re-attach process when a UP is detected as down.

Figure 25: N+2 UP Recovery Flow

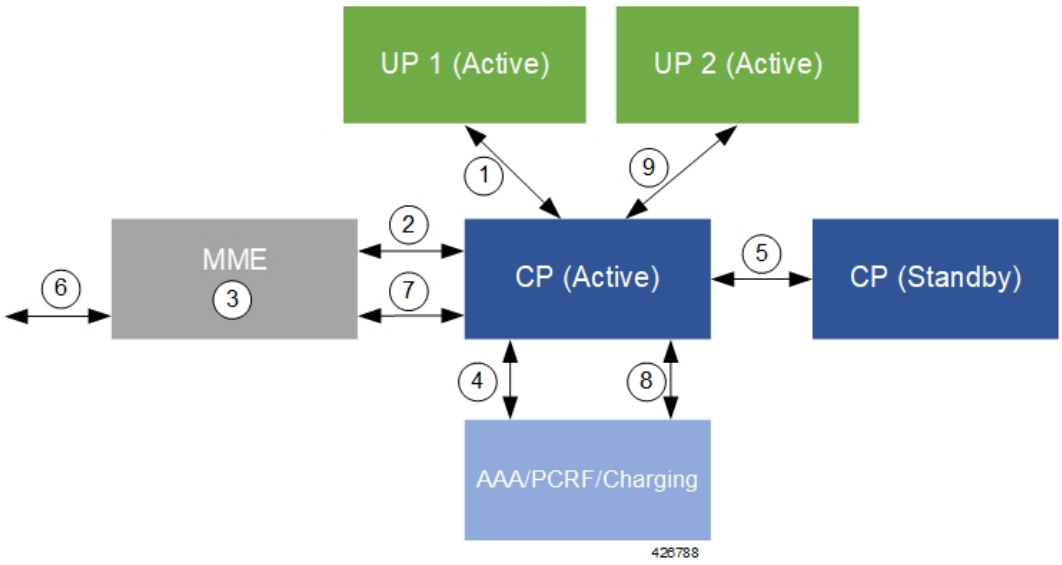


Table 19: N+2 UP Recovery Flow

Number	Description
1	The CP detects a UP failure.
2	The CP sends UP detach session messages to the MME(s) with a cause code of Local-Detach.
3	The MMEs process the request(s) and detach the sessions.
4	The CP communicates with the AAA/PCRF/Charging infrastructure to detach the sessions.
5	The CP (active) communicate with the standby CP to checkpoint the UP detach.

Number	Description
6	UEs whose sessions were previously detached, re-attach to the MME.
7	The MME communicates with the CP to re-attach UE sessions.
8	The CP communicates with the AAA/PCRF/Charging infrastructure to re-attach the sessions.
9	The CP completes the session re-attach process over the Sx interface with an alternate active UP.

Detailed detach and reattach on path failure flows for SAEGW CP/UP, P-GW CP/UP, S-GW CP/UP, and GnGp GGSN CP/UP are in the sections that follow.

Call Flows

SAEGW Detach and Reattach on Path Failure

The figure and the table that follows describe the detach and re-attach on path failure process for SAEGW CPs and UPs.

Figure 26: SAEGW CP/UP Detach and Re-attach on Path Failure Process

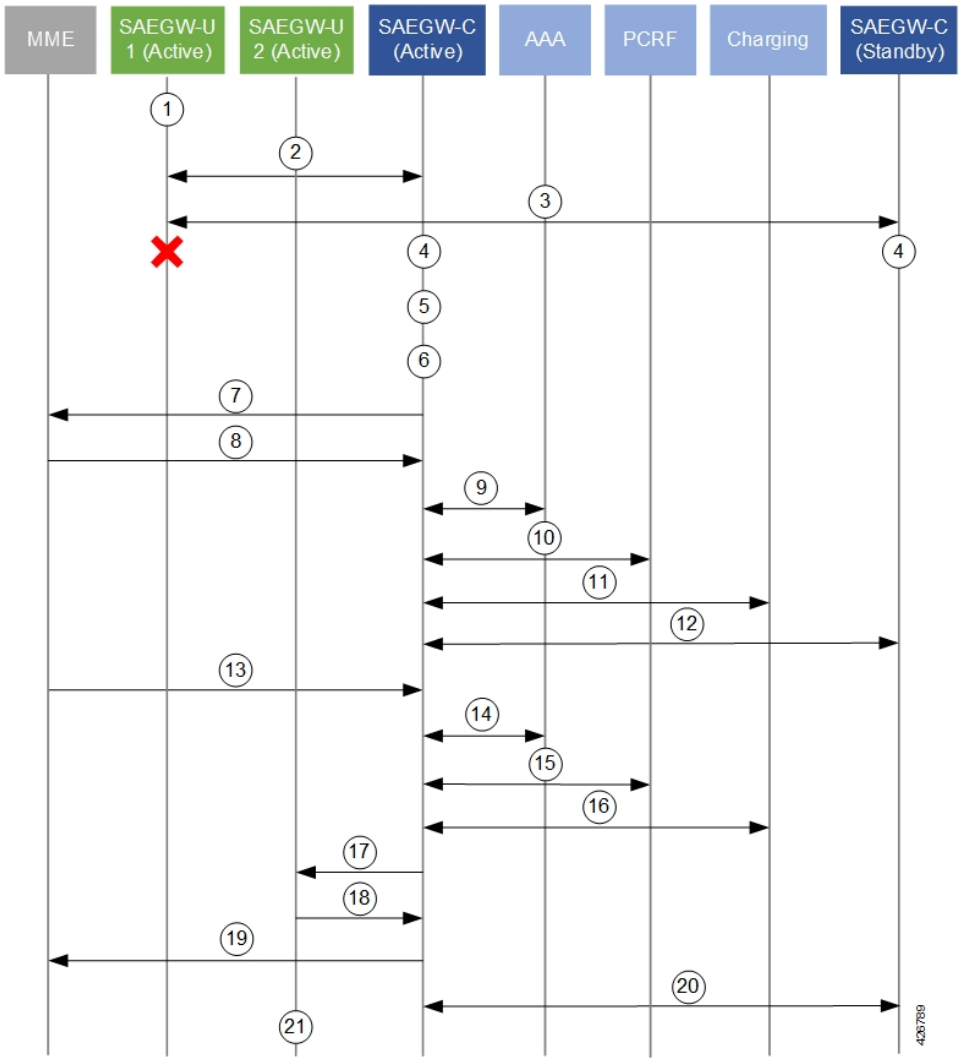


Table 20: SAEGW CP/UP Detach and Re-attach on Path Failure Process

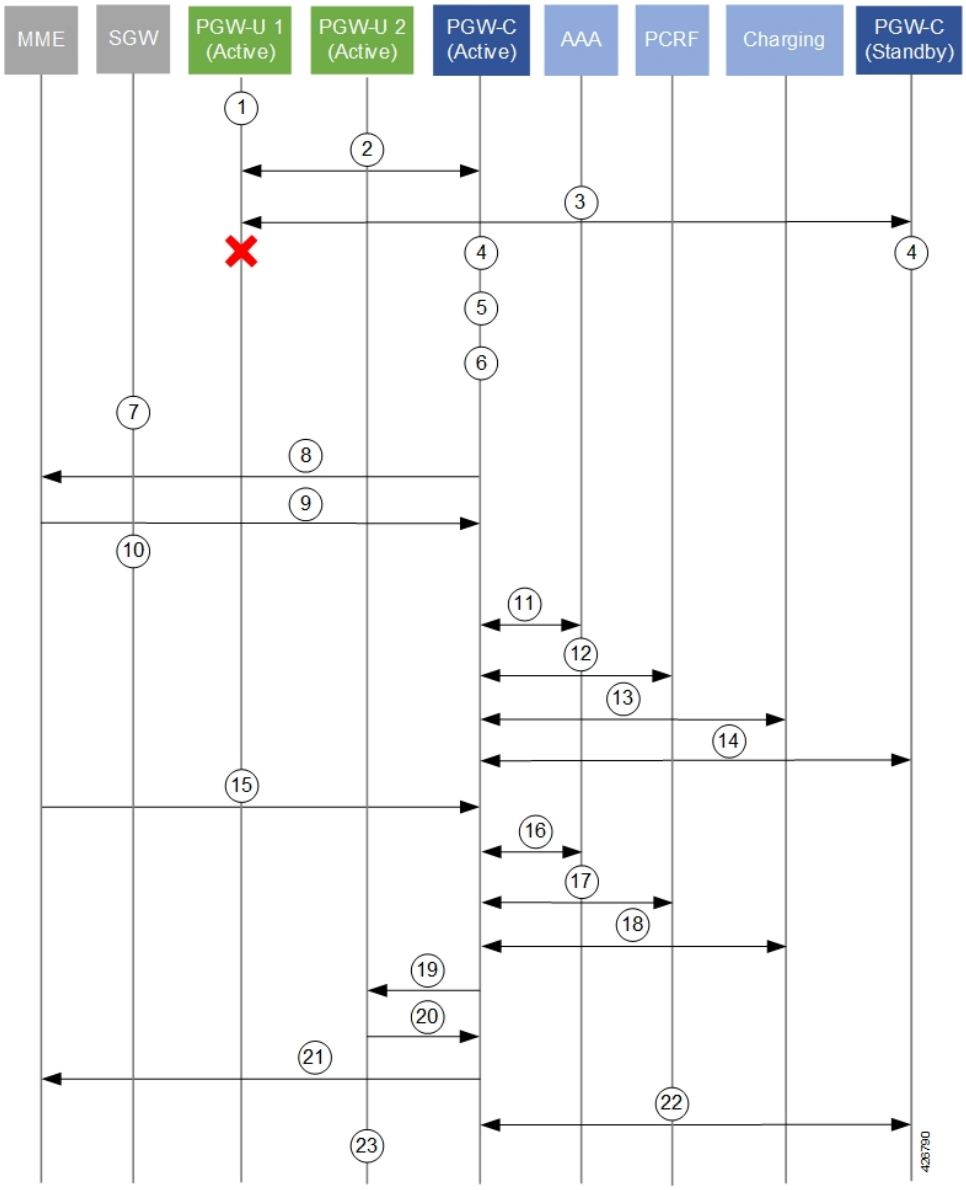
Number	Description
1	UE data sessions are processed by an active SAEGW UP.
2	The active SAEGW CP monitors SAEGW UPs via BFD and Sx-Heartbeat messages.
3	The secondary CP also monitors SAEGW UPs via BFD.
4	The active and standby CPs detect a BFD failure on a UP before eNB detection (relays on Sx timers (interval, retransmission, timeout)).
5	The BFD/VPNMGR on the active CP informs the Sx-demux process of a BFDDown event.
6	The Sx-demux process on the active CP initiates a path Failure notice to all Session Managers on the CP.

Number	Description
7	All Session Managers initiate the process of detaching sessions by sending Delete-bearer-req messages with a cause of Local-Detach to the MME. The detaches are initiated at a pre-defined rate.
8	The MME sends Delete-bearer-resp messages back to the CP. The MME does not page idle UEs with sessions being detached. The MME sends E-RAB release messages to active UEs with sessions being detached.
9	The active CP releases the session release with the AAA server(s).
10	The active CP releases the session with the PCRF.
11	The active CP releases the session with the Charging infrastructure.
12	The active CP syncs session detach information with the secondary CP.
13	For UEs re-initiating their session(s), the MME sends a Create-session-request message to the active CP. The MME selects the CP based on load algorithm (DNS, local config etc.).
14	The active CP processes the session attach request with the AAA server(s).
15	The active CP processes the session attach request with the PCRF.
16	The active CP processes the session attach request with the Charging infrastructure.
17	The active CP sends a Sx Session Establishment Request message to an alternate active UP. The CP selects the UP based on its load algorithm.
18	The UP sends a Sx Session Establishment Response message back to the CP.
19	The CP sends a Create-session-response message to the MME.
20	The active CP syncs information for the newly attached session with the secondary CP.
21	UE data sessions are now processed by the active SAEGW UP.

P-GW Detach and Reattach on Path Failure

The figure and the table that follows describe the detach and re-attach on path failure process for P-GW CPs and UPs.

Figure 27: P-GW CP/UP Detach and Re-attach on Path Failure Process



P-GW CP/UP Detach and Re-attach on Path Failure Process

Table 21: P-GW CP/UP Detach and Re-attach on Path Failure Process

Number	Description
1	UE data sessions are processed by an active P-GW UP.
2	The active P-GW CP monitors P-GW UPs via BFD and Sx-Heartbeat messages.
3	The secondary CP also monitors P-GW UPs via BFD.
4	The active and standby CPs detect a BFD failure on a UP before eNB detection (relays on Sx timers (interval, retransmission, timeout)).

Number	Description
5	The BFD/VPNMGR on the active CP informs the Sx-demux process of a BFDDown event.
6	The Sx-demux process on the active CP initiates a path Failure notice to all Session Managers on the CP.
7	The S-GW initiates a db-req to the MME.
8	All Session Managers initiate the process of detach sessions by sending Delete-bearer-req messages with a cause of Local-Detach to the MME. The detaches are initiated at a pre-defined rate.
9	The MME sends Delete-bearer-resp messages back to the CP. The MME does not page idle UEs with sessions being detached. The MME sends E-RAB release messages to active UEs with sessions being detached.
10	The S-GW forwards the db-resp to the PGW-C and removes it's PDN session.
11	The active CP releases the session release with the AAA server(s).
12	The active CP releases the session with the PCRF.
13	The active CP releases the session with the Charging infrastructure.
14	The active CP syncs session detach information with the secondary CP.
15	For UEs re-initiating their session(s), the MME sends a Create-session-request message to the active CP. The MME selects the CP based on load algorithm (DNS, local config etc.).
16	The active CP processes the session attach request with the AAA server(s).
17	The active CP processes the session attach request with the PCRF.
18	The active CP processes the session attach request with the Charging infrastructure.
19	The active CP sends a Sx Session Establishment Request message to an alternate active UP. The CP selects the UP based on its load algorithm.
20	The UP sends a Sx Session Establishment Response message back to the CP.
21	The CP sends a Create-session-response message to the MME.
22	The active CP syncs information for the newly attached session with the secondary CP.
23	UE data sessions are now processed by the active SAEGW UP.

S-GW Detach and Reattach on Path Failure

The figure and the table that follows describe the detach and re-attach on path failure process flow for S-GW CPs and UPs.

Figure 28: S-GW CP/UP Detach and Re-attach on Path Failure Process

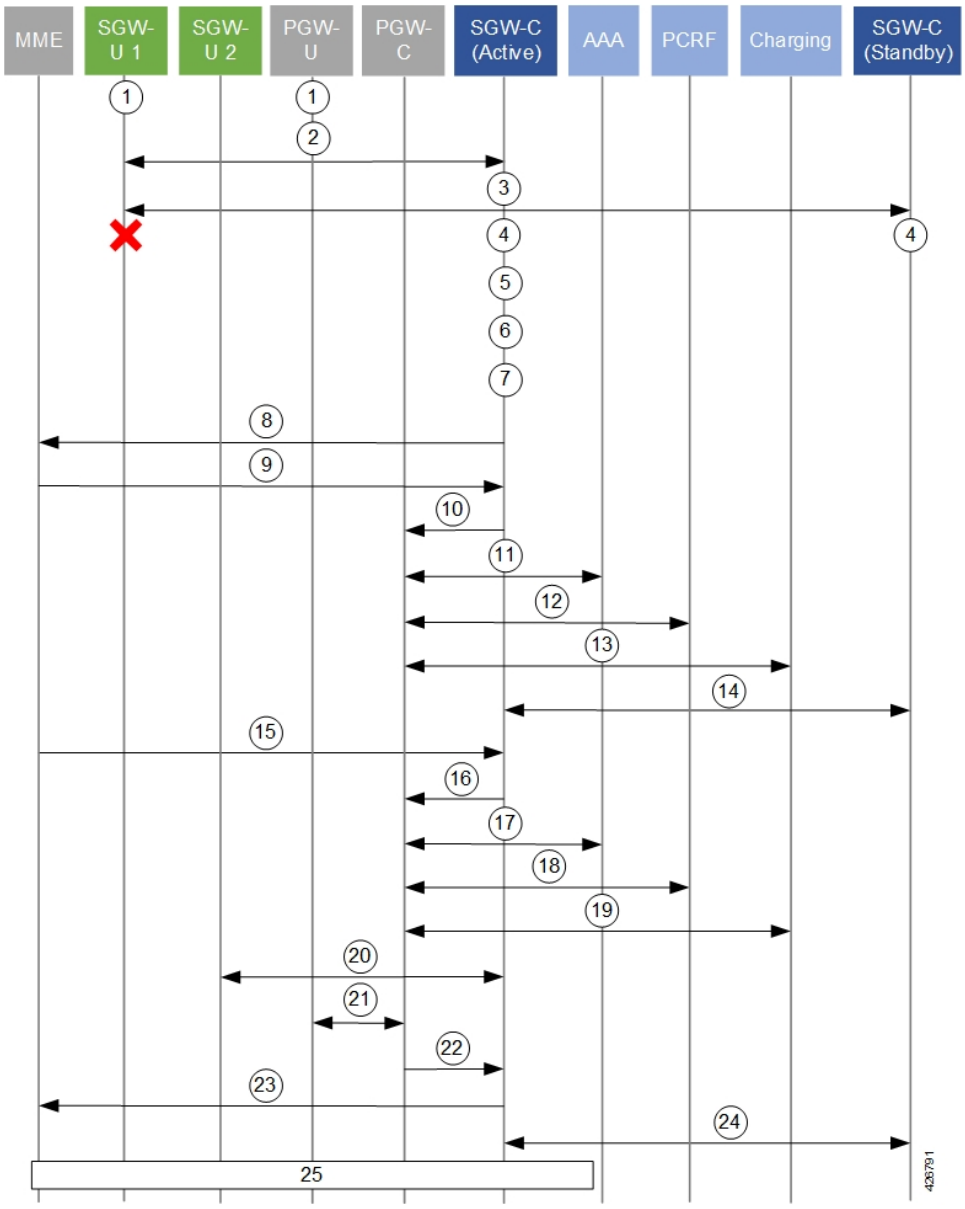


Table 22: S-GW CP/UP Detach and Re-attach on Path Failure Process

Number	Description
1	UE data sessions are processed by an active S-GW UP and an active PGW UP.
2	The active S-GW CP monitors S-GW UPs via BFD and Sx-Heartbeat messages.
3	The secondary S-GW CP also monitors S-GW UPs via BFD.
4	The active and standby S-GW CPs detect a BFD failure on the S-GW UP before eNB detection (relays on Sx timers (interval, retransmission, timeout)).

Number	Description
5	The BFD/VPNMGR on the active S-GW CP informs the Sx-demux process of a BFDDown event.
6	The Sx-demux process on the active CP initiates a path Failure notice to all Session Managers on the CP.
7	The S-GW CP initiates a db-req to the MME.
8	All Session Managers initiate the process of detach sessions by sending Delete-bearer-req messages with a cause of Local-Detach to the MME. The detaches are initiated at a pre-defined rate.
9	The MME sends Delete-bearer-resp messages back to the S-GW CP. The MME does not page idle UEs with sessions being detached. The MME sends E-RAB release messages to active UEs with sessions being detached.
10	The active S-GW CP releases the session release with the PGW UP.
11	The PGW CP releases the session with the AAA server(s).
12	The PGW CP releases the session with the PCRF.
13	The PGW CP releases the session with the Charging infrastructure.
14	The active S-GW CP syncs session detach information with the secondary S-GW CP.
15	For UEs re-initiating their session(s), the MME sends a Create-session-request message to the active S-GW CP. The MME selects the CP based on load algorithm (DNS, local config etc.).
16	The active S-GW CP relays the Create-session-request message to the PGW CP
17	The PGW CP processes the session attach request with the AAA server(s).
18	The PGW CP processes the session attach request with the PCRF.
19	The PGW CP processes the session attach request with the Charging infrastructure.
20	The active S-GW CP exchanges Sx Session Establishment Request and Response messages with an alternate active S-GW UP.
21	The active PGW CP exchanges Sx Session Establishment Request and Response messages with an active PGW UP.
22	The PGW CP sends a Create-session-response message to the S-GW CP.
23	The S-GW CP sends a Create-session-response message to the MME.
24	The active S-GW CP syncs information for the newly attached session with the secondary S-GW CP.
25	The S-GW CP and the complete the Modify Bearer Request procedure with the MME before UE data can flow through the active UPs.

GnGp GGSN Detach and Reattach on Path Failure

The figure and the table that follows describe the detach and re-attach on path failure process flow for GnGp GGSN CPs and UPs.

Figure 29: GnGp GGSN CP/UP Detach and Re-attach on Path Failure Process

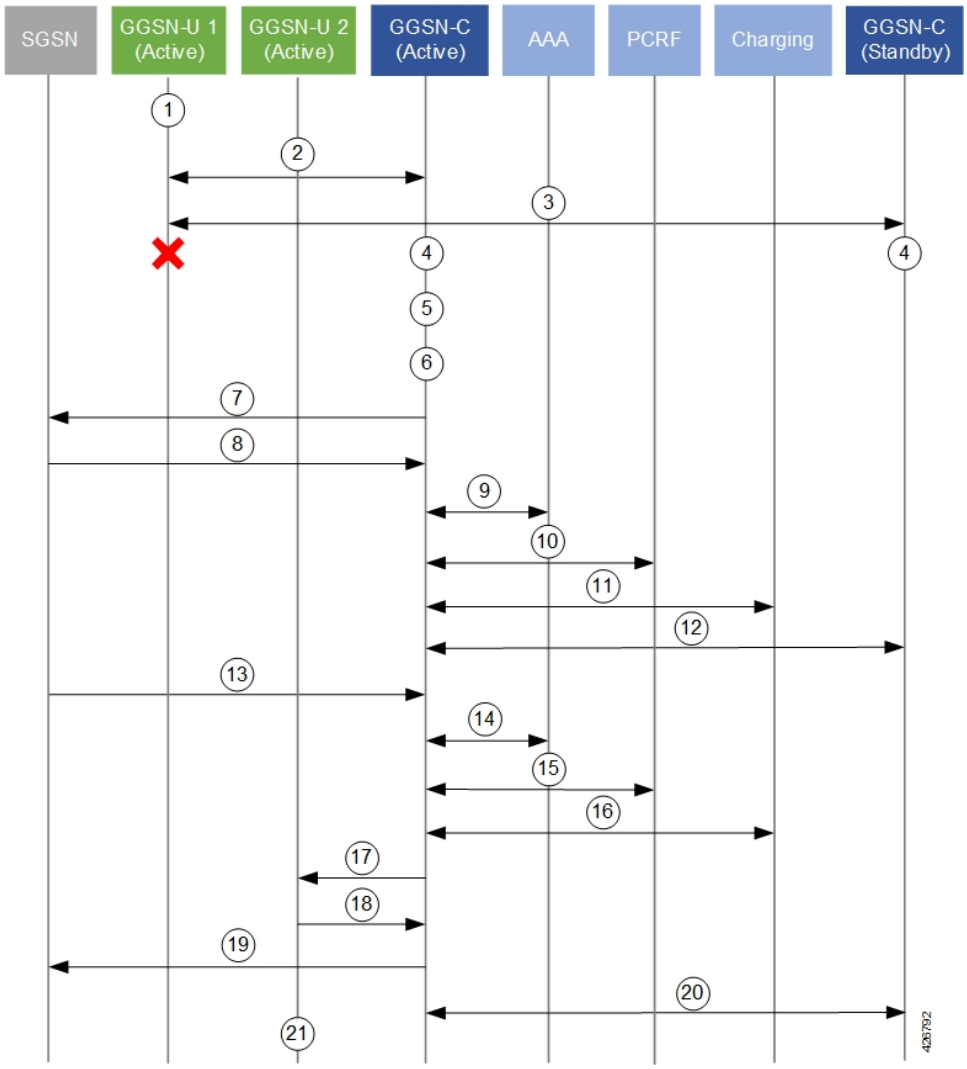


Table 23: GnGp GGSN CP/UP Detach and Re-attach on Path Failure Process

Number	Description
1	UE data sessions are processed by an active GGSN UP.
2	The active GGSN CP monitors GGSN UPs via BFD and Sx-Heartbeat messages.
3	The secondary CP also monitors GGSN UPs via BFD.
4	The active and standby CPs detect a BFD failure on a UP before eNB detection (relays on Sx timers (interval, retransmission, timeout)).
5	The BFD/VPNMGR on the active CP informs the Sx-demux process of a BFDDown event.
6	The Sx-demux process on the active CP initiates a path Failure notice to all Session Managers on the CP.

Number	Description
7	All Session Managers initiate the process of detaching sessions by sending Delete-pdp-context-req messages with no cause code to the SGSN. The detaches are initiated at a pre-defined rate.
8	The SGSN sends Delete-pdp-context-resp messages back to the CP. The SGSN does not page idle UEs with sessions being detached. The SGSN sends E-RAB release messages to active UEs with sessions being detached.
9	The active CP releases the session release with the AAA server(s).
10	The active CP releases the session with the PCRF.
11	The active CP releases the session with the Charging infrastructure.
12	The active CP syncs session detach information with the secondary CP.
13	For UEs re-initiating their session(s), the SGSN sends a Create-pdp-request message to the active CP. The SGSN selects the CP based on load algorithm (DNS, local config etc.).
14	The active CP processes the session attach request with the AAA server(s).
15	The active CP processes the session attach request with the PCRF.
16	The active CP processes the session attach request with the Charging infrastructure.
17	The active CP sends a Sx Session Establishment Request message to an alternate active UP. The CP selects the UP based on its load algorithm.
18	The UP sends a Sx Session Establishment Response message back to the CP.
19	The CP sends a Create-pdp-context response message to the SGSN.
20	The active CP syncs information for the newly attached session with the secondary CP.
21	UE data sessions are now processed by the active GGSN UP.

Additional N+2 Handling Scenarios

Beyond the flows described in the previous sections, the following table provides a description of network function (NF)/system behavior under various conditions with N+2 configured.

Table 24: N+2 Handling Scenarios

ID	Scenario	Handling	Notes
1	Active UP crash	Active CP detects BFD-failure with UP and detaches sessions belonging to that UP. Active CP propagates the disconnects to standby CP through SRP. When UP returns to active, it will re-associate with the active CP.	Detection occurs within the BFD timeout interval. CP Sx monitors BFD.
2	Active CP crash	Active CP switches over to standby CP. Active UP monitors Sx-heartbeat session for both active and standby CPs. Active UP does not purge sessions until ICSR failover time is reached.	Standby CP starts sending Sx-heartbeat upon failover – no sessions are purged by active UP.
3	Standby CP crash	Standby CP comes up and performs checkpoint with active CP to recover sessions	Sessions remain intact on active CP and active UP.
4	Network flaps between active CP and active UP; network between standby CP and active UP remains alive	Active CP detects BFD-Down for UP and initiates session detach processes and disassociates UP. Active CP propagates the disconnects to standby CP through SRP. Active UP monitors Sx-heartbeat with active CP. Active UP waits until configured Sx-heartbeat /path failure detection timeout occurs (>SRP switchover time) before clearing sessions.	

ID	Scenario	Handling	Notes
5	Network flaps between standby CP and active UP; active CP and active UP Sx-heartbeat also down	Active UP detects Sx-path failure. Active UP waits until configured Sx-heartbeat /path failure detection timeout occurs (>SRP switchover time) before clearing sessions. Active CP detects BFD-Down for UP and initiates session detach processes and disassociates UP.	UPs delete the sessions due to Sx-heartbeat timeout.
6	Network flaps between standby CP and active UP; Network between active CP and active UP is alive	Standby CP operates normally. Active CP-active is alive and responds to heartbeat. Active UP operates normally.	
7	Sx is not reachable, however BFD is reachable.	Active UP detects Sx-path failure. Active UP waits until configured Sx-heartbeat/path failure detection timeout occurs (>SRP switchover time) before clearing sessions. Active CP detects Sx-path failure for UP and initiates session detach processes and disassociates UP.	Corner case that is treated as Sx-path failure per current behavior (before N+2).
8	ICSR link between active and standby CPs goes down and standby CP also becomes active (Dual-Active case)	Upon becoming dual-Active, standby CP sends message to active UP with higher metric.	All service IPs advertised by dual-Active standby CP are with higher metric.
9	BGP failure Gn side of active UP	No action is taken in relation to N+2.	
10	BGP failure SGI side of active UP	No action is taken in relation to N+2.	
11	SessMgr crashes on active UP	Session recovery process occurs on active UP.	

ID	Scenario	Handling	Notes
12	Sx-demux crashes on active UP	Sx-demux recovery process occurs on active UP.	
13	VPP crashes on active UP	NPUMgr restarts the UP resulting in BFD loss triggering UP failure detection. Refer to Handling information for IDs 1 and 5 in this table.	
14	VPNMgr crashes on active UP	VPNMgr recovery process occurs on active UP.	
15	BFD crashes on active UP	BFD recovery process occurs on active UP.	
16	Sx-demux crashes on active CP	Sx-demux recovery process occurs on active CP. Sx-demux re-registers for BFD between CP and all UPs as part of recovery and rediscovers the state of each UP. Sx-demux recovers the restart-timestamp from the SessMgr.	It is possible for a UP state change to occur during the Sx-demux recovery on active CP (e.g. UP restarts but still shows as active to CP post recovery). Condition detected as follows: • Sx-demux recovers and CP detects either UP restart timestamp from Sx-heartbeat or UP-failure. • Based on this information, active CP initiates session purging.
17	VPNMgr crashes on active CP	VPNMgr recovery process occurs on active CP. BFDregistration information from recovered from SCT on active CP. Active CP restarts BFD with UP.	

ID	Scenario	Handling	Notes
18	BFD crashes on active CP	BFD recovery process occurs on active CP.	
19	SessMgr crashes on active CP	SessMgr recovery process occurs on active CP.	

Double Failure Handling Scenarios

N+2 double failure scenarios occur when there is a BFD failure followed by another event/failure. The handling of such scenarios is described in the following table.

Table 25: N+2 Double Failure Scenario Handling

ID	Scenario	Handling	Notes
1	Active CP fails while session detaches are in progress	ICSR switchover occurs between CPs. Standby CP becomes active CP. Active CP detects UP failure via BFD. Active CP detects UP restart vis Sx-heartbeat.	Impact: If UP restarts on double failure, it will have no sessions even though the standby CP will have recovered the sessions. These sessions are then cleaned as part of session replacement or session disconnects from UEs. If UP does not restart then the CP-new-active clears the sessions of the failed UP.
2	Standby CP fails while session detaches are in progress	Standby CP checkpoints state information with the Active CP. Information pertaining to deleted sessions is invalidated from active CP.	
3	Active CP determines UP failure due to router flap; Active CP receives UP BFD after initiating session detaching	Once UP BFD down is initially detected, all sessions are detached.	

BFD Flapping and VPC

N+2 uses BFD to monitor the existence/viability of a network path between the session endpoints. By using multihop BFD with loopback endpoints, the BFD session state functions as a proxy for the state of the system to which it connects.

However, a BFD session can go down, or bounce/flap, for reasons other than far-side system failure (e.g. due to ARP storms or router misconfiguration). If the disruption is sufficiently severe and long lasting, it can cause systems on both sides to detect BFD session failure even though both systems are functional.

Configuration adjustments can be made to help offset the occurrence of such events.

The following recommendations are offered based on the platform on which your NFs are deployed:

- VPC-SI: Adjust the BFD multihop-peer settings to increase the BFD detection time to 2-3 sec and the number of retries correspondingly.
- VPC-DI: CF switchover and SF migration can interrupt BFD packet generation and processing for multiple seconds. To prevent BFD session flaps when these events occur, BFD detection time for sessions involving VPC-DI systems must be set to 7 seconds or longer.

Sx-association Scenarios

The following table provides information on associating and disassociating CPs and UPs when using N+2.

Table 26: N+2 Sx-association Scenarios

Scenario	Mechanism(s)
Sx-disassociation from UP to CP	• Sx-demux to disable BFD monitoring with VPNMgr • SAEGW-service is removed • Sx-disassociation from UP
Adding UPs	As part of Day-0: • Add BFD loopback address for UP. • Configurate BFD on CPs. • Add UP Group and configure it for selection on CPs.
Removing UPs	On CP, execute the CLI command to clear subscribers with IP address of UP and keyword to block new sessions being placed on that UP. • Verify that all the subscribers are torn down on UP. • On the UP, execute the CLI command to disassociate from CP. This will disassociate the UP from CP and CP will not choose this UP for further sessions. Verify that all the sessions have been torn down. • On CP, remove the UP from the UP Group. • On CP, execute the CLI command to remove the UP from the UP Group (this will also deregister the BFD monitoring of the UP). • Disable the BFD configurations for monitoring at UP and at CP: no monitor-group CLI command.
UP-initiated Sx-association	Sx-demux on CP starts processing the BFDUp and BFDDown notifications from VPNMGr.

UP-released Sx-association	Sx-demux on CP ignores the BFDUp and BFDDown notifications from VPNMgr.

N+2 and IP Addressing

Loopback IP Addresses

The following is true of BFD loopback addresses in relation to N+2:

- BFD loopback-IP-Address on the active CP and standby CP must be configured on Day-0.
- BFD operates between the active CP and active UP as well as between the standby CP and active UP. As such, all three components must use unique BFD loopback-IP-addresses
- For each CP and UP, configured BFD loopback-IP-addresses must be different from the addresses used for the Sx interfaces, and, in the case of the CPs must also be different from the addresses used for the SRP interface.

IP Address Availability

With the N+2 deployment scenario, UEs may re-attach at a high rate (comparable to the detach rate). To facilitate this process, UPs must have sufficient IP addresses available.

CUPS IP Pool Management includes the capability to provision UPs with "chunks" of addresses. The chunk size and number of pools configured on the CP need to be increased proportionately so as to accommodate the high rate of re-attachments from the CP to UP such that sessions do not get rejected by the UP due to unavailability of IP addresses.

The potential re-attach rate can be roughly estimated by multiplying the number of Session Manager tasks processing UP sessions by 1000 sessions/second.

Address capacity is determined by multiplying the size of the chunk (between 16 and 8192) and the number of IP pools. Both configured on the CP.

Configuring N+2 UP Recovery

To configure N+2 UP Recovery:

1. Configure BFD on the CP and UP.

```
configure
   context bfd_context_name
      ip route static multihop bfd mhbfd_session_name local_endpoint_ip_address
 remote_endpoint_ip_address
      bfd-protocol
         bfd multihop-peer dst_ip_address interval tx_interval min_rx
rx_interval multiplier value
         #exit
      #exit
```

NOTES:

- *bfd_ctx_name* is the name of the context in which BFD is to be configured. This must be the same context in which Sx is configured.
- *mhbfd_session_name* is a name for the BFD session route. Multiple session routes can be created, one for each peer connection.
- *local_endpoint_ip_address* is the IPv4 or IPv6 address corresponding to the local interface in the current context.
- *remote_endpoint_ip_address* is the IPv4 or IPv6 address corresponding to the remote BFD peer.
  - If this route is being configured on the CP, then the remote address is that of the peer UP.
  - If this route is being configured on the UP, then the remote address is that of the peer CP.
- *dst_ip_address* is the IPv4 or IPv6 address corresponding to the remote BFD peer. This must be the same as the *remote_endpoint_ip_address* interface configured for the static multihop BFD route. Multiple peers can be configured, one for each remote peer.
- **interval** *tx_interval* is the transmit interval (in milliseconds) between BFD packets.
- **min_rx** *rx_interval* is the minimum receive interval capability (in milliseconds) between BFD packets.
- **multiplier** *value* the multiplier value used to compute holddown.
- To determine the Detect Time (X), you can use the following calculation:

  Detect Time (X) = **interval** *tx_interval* * **multiplier** *value*

  The recommended value of Detect time (X) is 3 seconds for VPC-SI, and 7 seconds for VPC-DI.

2. Configure the BFD-loopback per context on the CP and UP.

```
configure
   context monitor_ctx_name
      monitor-protocols
         monitor-group monitor_group_name protocol bfd
           session-ctx session_ctx_name local-addr { ipv4_address | ipv6_address
 } remote-address { ipv4_address | ipv6_address }
             #exit
```

NOTES:

- *Monitor_ctx_name* is the name of the context in which BFD monitoring is to be configured. This must be the same context in which Sx is configured.
- *Monitor_group_name* is the name of the group specifying the BFD monitoring parameters. Multiple monitor-groups can be configured.
- *Session_ctx_name* is the name of the context containing the local interfaces over which BFD monitoring will occur. This must be the same context in which Sx is configured.
- **local-addr {** *ipv4_address* **|** *ipv6_address* **}** is the IPv4 or IPv6 address corresponding to the local interface in the specified context.
- **remote-addr {** *ipv4_address* **|** *ipv6_address* **}** is the IPv4 or IPv6 address corresponding to the remote peer with which BFD monitoring will occur.

- If this monitor group is being configured on the CP, then the remote address is that of the UP group.
- If this monitor group is being configured on the UP, then the remote address is that of the CP.

3. Configure the BFD-loopback (remote-IP) within a specific UP-group on the CP:

```
configure
   user-plane-group up_group_name
      peer-node-id { ipv4_address | ipv6_address } monitor-group-name
monitor_group_name
      #exit
```

NOTES:

- *up_group_name* is the name of the UP group containing the data UPs for N+2 UP Recovery will be supported.
  - This cannot be the default group.
  - This group should not contain UPs intended to support IMS/VoLTE.
- **{** *ipv4_address* **|** *ipv6_address* **}** is the IPv4 or IPv6 address of the Sx interface on an active UP that will be part of the UP group. Multiple peer-nodes can be configured within the group. Note that the Sx interface is a different interface from the one that will be used to monitor BFD.
- *monitor_group_name* is the name of the monitoring group the UP will be associated with.

Monitoring and Troubleshooting

Show Commands

show sx peers { full address *peer_ip_address* **| wide }**

show sx peers full address *peer_ip_address*

Displays the Monitor-related information for the specified peer (e.g. VPN context name, group name, and state).

show sx peers wide

Displays "Monitor State" with the default state being "U" for UP, "D" for Down, and "N" for Not Applicable.

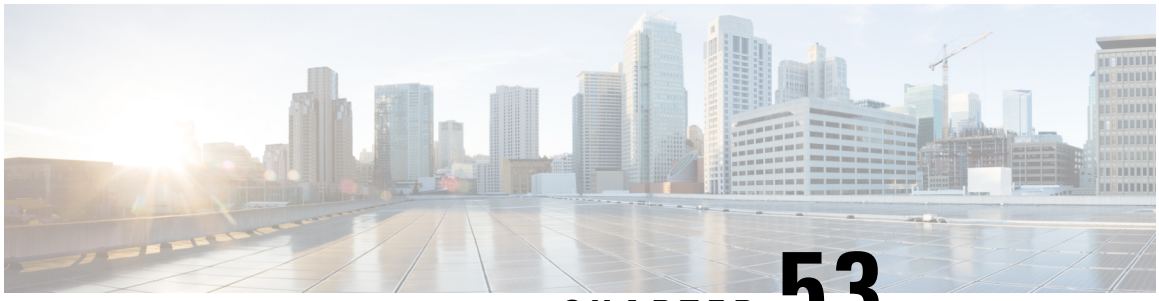
show sx-service statistics all

SNMP

The following SNMP traps can be used to monitor N+2 UP Recovery health:

- starBFDSessUp (starentTraps 1276)
- starBFDSessDown (starentTraps 1277)

- starSxPathFailure (starentTraps 1382) – This trap has been updated to include a new cause code: bfd-failure(8)
- starSxPathFailureClear (starentTraps 1383)



CHAPTER 53

NAT Support

- Feature Summary and Revision History, on page 437
- Feature Description, on page 437
- Configuring NAT in CUPS, on page 439
- Monitoring and Troubleshooting, on page 440

Feature Summary and Revision History

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced.	Pre 21.24

Feature Description

CUPS supports Network Address Translation (NAT) which allows you to configure network addresses. The system can be configured to automatically forward data packets encapsulating the source IP or Source port address of the UE with NAT IP address and NAT port.

The supported NAT combinations include:

- NAT44 On Demand Many to One
- NAT44 On Demand One to One
- NAT64 On Demand Many to One
- NAT 64 On Demand One to One
- NAT44 Not On Demand Many to One

- NAT44 Not On Demand One to One
- NAT64 Not On Demand Many to One
- NAT64 Not On Demand One to One

For supplemental information about NAT, see StarOS *NAT Administration Guide*.

NOTE: Not all features and/or functionality that are mentioned in StarOS *NAT Administration Guide* are applicable in the CUPS architecture.

Limitations

NAT support has the following limitations:

- Only NAT44 with many-to-one and on-demand mode is supported.
- All NAT pools are configured at respective User Plane on destination context.
- Charging action with CLI action deny in fw-and-nat policy and for flow-any-error charging action in active-charging-service is not supported.
- Access-rules which are configured with "dynamic-only" and "static-and-dynamic" – rules from external servers are not supported.
- Multiple IP support from same realm is not supported with this feature.
- Next hop forwarding in NAT pool is not supported.
- Port range in NAT pool is not supported.
- Skip private IP check CLI is not supported.
- RADIUS and Gy returned Fw-and-nat policy-based applying NAT policy is not supported.
- Bearer specific filters are not supported in access-ruledefs.
- Access-rules do not support trigger open-port port range config in fw-and-nat policy.
- NAT port recovery (fw-and-nat action) is not supported after SR/ICSR.
- NAT Re-assembly Timeout CLI is not supported in active-charging service. The generic context level CLI on UP must be used instead.
- NAT fragmentation re-assembly failure is not supported due to open bugs in basic CUPS re-assembly.
- NAT flow-mapping timer is not supported
- For N:M redundancy, the NAT IP pools to be configured from RCM done as part of interface config for each UP host and the pool name needs to be unique across all the active User Planes. This makes it mandatory to use NAT Groups for all the pools so that the same NAT realm referred in fw-and-nat policy can be applicable for all the User Planes.
- In case of N:M redundancy, the total number of NAT IP pools collectively configured on all UPs via RCM must be as per the maximum limit (2000) of IP pools. The configuration in standby User Plane fails if the cumulative total of all active UPs exceeds the maximum value.

Configuring NAT in CUPS

The relevant configuration of NAT is done at CP and pushed to UP. Only pool-related configuration is present on User Plane.

For information on NAT-related CLI commands, refer to the *StarOS NAT Administration Guide > NAT Configuration* chapter.

NOTE: Not all CLI commands and configurations mentioned in the *StarOS NAT Administration Guide > NAT Configuration* chapter are applicable in CUPS architecture.

Sample Configurations

Control Plane

The following is a sample configuration required at Control Plane for enabling NAT in CUPS. This configuration is pushed to User Plane during User Plane registration through PFD mechanism.

```
configure
active-charging service ACS
     access-ruledef all
       ip any-match = TRUE
     #exit
     access-ruledef udp
       udp any-match = TRUE
     #exit
     access-ruledef tcp
       tcp any-match = TRUE
     #exit
     access-ruledef icmp
       icmp any-match = TRUE
     #exit
     fw-and-nat policy NatPolicy1
        access-rule priority 1 access-ruledef tcp permit nat-realm NAT44_GRP1
        access-rule priority 2 access-ruledef icmp permit nat-realm NAT44_GRP1
        #access-rule priority 2 access-ruledef r2 permit bypass-nat
        nat policy ipv4-only default-nat-realm NAT44_PUBLIC5
        nat binding-record edr-format NBR port-chunk-allocation port-chunk-release
     #exit

     fw-and-nat policy NatPolicy2
        access-rule priority 1 access-ruledef all permit nat-realm NAT44_PUBLIC1
        #access-rule priority 2 access-ruledef r2 permit bypass-nat
        nat policy ipv4-only
        nat binding-record edr-format NBR port-chunk-allocation port-chunk-release
     #exit

     rulebase cisco
     fw-and-nat default-policy NatPolicy1
     flow end-condition normal-end-signaling session-end timeout edr NBR
     #exit
#exit
end
```

User Plane

The following pool-related configuration is required at User Plane in ISP context.

```
configure
   context ISP1-UP
      ip pool NAT44_PUBLIC1 209.165.200.225 255.255.255.224 napt-users-per-ip-address 2
on-demand port-chunk-size 16 max-chunks-per-user 4 group-name NAT44_GRP1
      ip pool NAT44_PUBLIC2 209.165.200.226 255.255.255.224 napt-users-per-ip-address 2
on-demand port-chunk-size 16 max-chunks-per-user 4 group-name NAT44_GRP1
      ip pool NAT44_PUBLIC3 209.165.200.227 255.255.255.224 napt-users-per-ip-address 2
on-demand port-chunk-size 8 max-chunks-per-user 1 group-name NAT44_GRP2
      ip pool NAT44_PUBLIC4 209.165.200.228 255.255.255.224 napt-users-per-ip-address 4
on-demand port-chunk-size 32256 max-chunks-per-user 4 group-name NAT44_GRP2
      ip pool NAT44_PUBLIC5 209.165.200.229 255.255.255.224 napt-users-per-ip-address 8064
 on-demand port-chunk-size 8 max-chunks-per-user 2
      end
```

Sample NAT Pool Related Configuration for Different NAT Pool Types

```
1-1 on-demand:
----------------------------
config
context ISP1-UP
ip pool NAT44_ipv4_1_1 209.165.200.230 255.255.255.224 nat-one-to-one on-demand
nat-binding-timer 60
end

N-1 Not-on-demand:
------------------------
config
context ISP1-UP
ip pool NAT44_ipv4_N_1 209.165.200.231 255.255.255.224 napt-users-per-ip-address 2
max-chunks-per-user 2 port-chunk-size 8
end

1-1 Not-on-demand:
------------------------
config
context ISP1-UP
ip pool NAT44_ipv4_NOD_1_1 209.165.200.232 255.255.255.224 nat-one-to-one
end
```

Note In Control Plane configuration needs to be added along with one or more access ruledef mapped to any of the required NAT Pool/Group configured in User Plane. For more information, see *Ultra Packet Core CUPS Control Plane Administration Guide*.

Monitoring and Troubleshooting

Gathering NAT Statistics in CUPS

The following table lists the commands that can be used to gather NAT statistics. The first column lists what statistics to gather and the second column lists the user plane command to use.

Statistics/Information	Show Command
Information for all current subscribers who have either active or dormant sessions. Checks IP address associated with subscriber. Also displays all the IP addresses that are in use in a NAT realm.	**show subscribers user-plane-only full all**
Information on NAT subsystem statistics.	**show user-plane-service statistics all**
All NAT-related statistics.	**show user-plane-service statistics nat all**
All NAT Realm-related statistics.	**show user-plane-service statistics nat nat-realm all**
Statistics of all NAT IP pools in a NAT IP pool group.	**show user-plane-service statistics nat nat-realm** *pool_name*
Information on NAT bind records generated.	**show user-plane-service edr-format statistics all**
Verifying association of fw-and-nat policy in APN on UP.	**show user-plane-service pdn-instance name** *name*
Verifying cofiguration of fw-an-nat policy on UP.	**show user-plane-service fw-and-nat policy all**
Information on NAT bind records generated for port chunk allocation and release.	**show user-plane-service rulebase name** *name*
Information on access ruledef.	**show user-plane-service ruledef all**
Verifying association of fw-and-nat policy in rulebase on UP.	**show user-plane-service rulebase name** *name*

Clear Commands

The following clear CLI commands are available in support of this feature:

- **clear user-plane-service statistics nat nat-realm all**
- **clear user-plane-service statistics nat all**

SNMP Traps for NAT Parameter Thresholds

The following SNMP traps for NAT parameter thresholds are supported in CUPS.

SNMP Traps	Description
ThreshNATPortChunks	Generated when NAT port chunk usage reaches configured threshold limit
ThreshClearNATPortChunks	Generated when NAT port chunk usage reaches configured clear threshold limit.

SNMP Traps	Description
ThreshNATPktDrop	Generated when NAT packet drop reaches configured threshold limit.
ThreshClearNATPktDrop	Generated when NAT packet drop reaches configured clear threshold limit.
ThreshIPPoolUsed	Generated when the number of IPs used in the IP Pool reaches configured threshold limit.
ThreshClearIPPoolUsed	Generated when the number of IPs used in the IP Pool reaches configured clear threshold limit.
ThreshIPPoolFree	Generated when IP pool is free and threshold limit reached.
ThreshClearIPPoolFree	Generated when IP pool is used, and clear threshold limit reached.
ThreshIPPoolAvail	Generated when IP pool is available for next flow and configured threshold reached.
ThreshClearIPPoolAvail	Generated when IP pool is used, and configured threshold is reached.

NOTE: The respective CLIs must be configured in the User Plane to enable these traps.

Bulk Statistics

Context Schema

Table 27: Context Schema

Variable Name	Data Type	Counter Type	Description
nat-total-flows	Int64	Counter	Total number of NAT44 and NAT64 flows
nat44-total-flows	Int64	Counter	Total number of NAT44 flows
nat64-total-flows	Int64	Counter	Total number of NAT64 flows
bypass-nat-total-flows	Int64	Counter	Total number of NAT44 and NAT64 Bypass NAT flows
bypass-nat-ipv4-total-flows	Int64	Counter	Total number of NAT44 Bypass NAT flows
bypass-nat-ipv6-total-flows	Int64	Counter	Total number of NAT64 Bypass NAT flows
nat-current-flows	Int64	Gauge	Current number of NAT44 and NAT64 flows
nat44-current-flows	Int64	Gauge	Current number of NAT44 flows
nat64-current-flows	Int64	Gauge	Current number of NAT64 flows
bypass-nat-current-flows	Int64	Gauge	Current number of NAT44 and NAT64 Bypass NAT flows

Variable Name	Data Type	Counter Type	Description
bypass-nat-ipv4-current-flows	Int64	Gauge	Current number of NAT44 Bypass NAT flows
bypass-nat-ipv6-current-flows	Int64	Gauge	Current number of NAT64 Bypass NAT flows
sfw-total-rxpackets	Int64	Counter	Total number of packets received by the Service
sfw-total-rxbytes	Int64	Counter	Total number of bytes received by the Service
sfw-total-txpackets	Int64	Counter	Total number of packets transferred by the Service
sfw-total-txbytes	Int64	Counter	Total number of bytes transferred by the Service
sfw-total-injectedpkts	Int64	Counter	Total number of packets injected by the Service
sfw-total-injectedbytes	Int64	Counter	Total number of bytes injected by the Service
sfw-dnlnk-droppkts	Int64	Counter	Total number of downlink packets dropped by the Service
sfw-dnlnk-dropbytes	Int64	Counter	Total number of downlink bytes dropped by the Service
sfw-uplnk-droppkts	Int64	Counter	Total number of uplink packets dropped by the Service
sfw-uplnk-dropbytes	Int64	Counter	Total number of uplink bytes dropped by the Service

Note Schema is supported in User Plane for CUPS.

ECS Schema

Table 28: ECS Schema

Variable Name	Data Type	Counter Type	Description
nat-current-ipv4-pdn-subscribers	Int32	Gauge	Current number of NAT IPv4 PDN Subscribers
nat-current-ipv6-pdn-subscribers	Int32	Gauge	Current number of NAT IPv6 PDN Subscribers
nat-current-ipv4v6-pdn-subscribers	Int32	Gauge	Current number of NAT IPv4v6 PDN Subscribers
nat-total-ipv4-pdn-subscribers	Int64	Counter	Total number of NAT IPv4 PDN Subscribers
nat-total-ipv6-pdn-subscribers	Int64	Counter	Total number of NAT IPv6 PDN Subscribers
nat-total-ipv4v6-pdn-subscribers	Int64	Counter	Total number of NAT IPv4v6 PDN Subscribers

Variable Name	Data Type	Counter Type	Description
nat-current-ipv4-pdn-subscribers-with-nat-ip	Int32	Gauge	Current number of NAT IPv4 PDN Subscribers with NAT IP
nat-current-ipv6-pdn-subscribers-with-nat-ip	Int32	Gauge	Current number of NAT IPv6 PDN Subscribers with NAT IP
nat-current-ipv4v6-pdn-subscribers-with-nat-ip	Int32	Gauge	Current number of NAT IPv4v6 PDN Subscribers with NAT IP
nat-total-ipv4-pdn-subscribers-with-nat-ip	Int64	Counter	Total number of NAT IPv4 PDN Subscribers with NAT IP
nat-total-ipv6-pdn-subscribers-with-nat-ip	Int64	Counter	Total number of NAT IPv6 PDN Subscribers with NAT IP
nat-total-ipv4v6-pdn-subscribers-with-nat-ip	Int64	Counter	Total number of NAT IPv4v6 PDN Subscribers with NAT IP
nat-total-unsolicited-dwnlnk-pkts	Int64	Counter	Total number of unslolicited downlink packets received
nat-total-icmp-hu-sent-for-dwnlnk-pkts	Int64	Counter	Total number of ICMP host unreachable sent for downlink packets

Note Schema is supported in User Plane for CUPS.

NAT-realm Schema

The NAT realms are configured in User Plane and statistics are stored per-context per-realm. These statistic variables, both cumulative and snapshot, are available in the nat-realm schema.

Table 29: NAT-realm Schema

Variable Name	Data Type	Counter Type	Description
Vpnname	String	Info	Context name.
Realmname	String	Info	Realm name.
nat-rlm-bind-updates	Int64	Counter	Total interim AAA NBU sent.
nat-rlm-bytes-txferred	Int64	Counter	Total number of NAT44 and NAT64 bytes transferred by realm (uplink + downlink).
nat-rlm-bytes-nat44-tx	Int64	Counter	Total number of NAT44 bytes transferred by realm.
nat-rlm-bytes-nat64-tx	Int64	Counter	Total number of NAT64 bytes transferred by realm.

Variable Name	Data Type	Counter Type	Description
nat-rlm-ip-flows	Int64	Counter	Total number of NAT44 and NAT64 flows used by the realm.
nat-rlm-nat44-flows	Int64	Counter	Total number of NAT44 flows processed by realm.
nat-rlm-nat64-flows	Int64	Counter	Total number of NAT64 flows processed by realm.
nat-rlm-ip-denied	Int32	Counter	Total number of NAT44 and NAT64 flows denied NAT IP address.
nat-rlm-ip-denied-nat44	Int64	Counter	Total number of NAT44 flows denied IP.
nat-rlm-ip-denied-nat64	Int64	Counter	Total number of NAT64 flows denied IP.
nat-rlm-port-denied	Int32	Counter	Total number of NAT44 and NAT64 flows denied ports.
nat-rlm-port-denied-nat44	Int64	Counter	Total number of NAT44 flows denied ports.
nat-rlm-port-denied-nat64	Int64	Counter	Total number of NAT64 flows denied ports.
nat-rlm-memory-denied	Int64	Counter	Total number of NAT44 and NAT64 flows denied memory.
nat-rlm-memory-denied-nat44	Int64	Counter	Total number of NAT44 flows denied memory.
nat-rlm-memory-denied-nat64	Int64	Counter	Total number of NAT64 flows denied memory.
nat-rlm-ttl-ips	Int32	Gauge	Total number of NAT public IP addresses, per context per NAT realm. Is a static value.
nat-rlm-ips-in-use	Int32	Gauge	Total number of NAT IP addresses currently in use, per context per NAT realm.
nat-rlm-current-users	Int32	Gauge	Total number of subscribers currently using the NAT realm.
nat-rlm-ttl-port-chunks	Int32	Gauge	Total number port-chunks, per context per NAT realm. Is a static value.
nat-rlm-chunks-in-use	Int32	Gauge	Total number of port-chunks currently in use, per context per NAT realm.
nat-rlm-port-chunk-size	Int32	Gauge	Size of the port chunk in the NAT realm.
nat-rlm-port-chunk-average-usage-tcp	Int32	Gauge	Average TCP port usage in the allocated TCP ports, i.e. out of allocated TCP ports how many got used. Not percentage value.

Variable Name	Data Type	Counter Type	Description
nat-rlm-port-chunk-average-usage-udp	Int32	Gauge	Average UDP port usage in the allocated UDP ports, i.e. out of allocated UDP ports how many got used. Not percentage value.
nat-rlm-port-chunk-average-usage-others	Int32	Gauge	Average other (ICMP or GRE) port usage in the allocated other ports, i.e. out of allocated 'other' ports how many got used. Not percentage value.
nat-rlm-max-port-chunk-subs	Int64	Counter	Total number of subscribers who used maximum number of port chunks.
nat-rlm-max-port-chunk-used	Int32	Counter	Maximum port chunks used.
nat-rlm-max-cur-port-chunk-subs	Int64	Gauge	Current number of subscribers using maximum number of port chunks.
nat-rlm-max-cur-port-chunk-used	Int32	Gauge	Maximum port chunks used by active subscribers.

EDRs

The following NAT-specific attributes are supported in regular EDRs:

- sn-nat-subscribers-per-ip-address: Subscriber(s) per NAT IP address
- sn-subscriber-nat-flow-ip: NAT IP address of NAT-enabled subscribers
- sn-subscriber-nat-flow-port: NAT port number of NAT-enabled subscribers

Sample EDR

```
#sn-start-time,sn-end-time,ip-protocol,ip-subscriber-ip-address,ip-server-ip-address,sn-subscriber-port,sn-server-port,
sn-nat-ip,sn-nat-port-block-start,sn-nat-port-block-end,sn-subscriber-nat-flow-ip,sn-subscriber-nat-flow-port,sn-nat-realm-name,
sn-nat-subscribers-per-ip-address,sn-nat-binding-timer,sn-nat-gmt-offset,sn-nat-port-chunk-alloc-dealloc-flag,sn-nat-port-chunk-alloc-time-gmt,
sn-nat-port-chunk-dealloc-time-gmt,sn-nat-no-port-packet-dropped,sn-closure-reason
02/18/2020 12:11:11:630,02/18/2020
12:11:11:632,1,209.165.200.225,209.165.201.1,0,0,,,,209.165.200.230,1024,,2,,,,,,0,0
02/18/2020 12:11:08:672,02/18/2020
12:11:09:671,6,209.165.200.225,209.165.201.1,1001,3000,,,,209.165.200.230,1034,,2,,,,,,0,0
02/18/2020 12:11:14:499,02/18/2020
12:11:14:499,17,209.165.200.225,209.165.201.1,1001,3000,,,,209.165.200.240,1025,,8064,,,,,,0,0
```

NAT Binding Records

Whenever a NAT IP address or NAT port-chunk is allocated/deallocated to/from a subscriber, NAT Binding Records (NBR) can be generated. Generation of NBRs is configurable in the Firewall-and-NAT policy configuration.

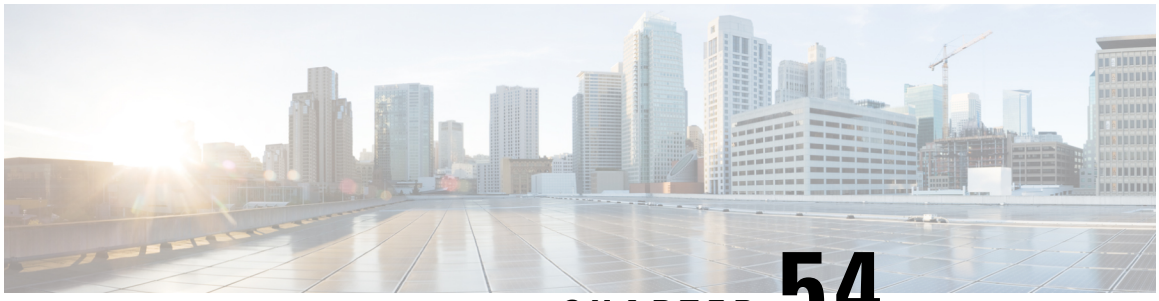
Sample NBR

```
#sn-start-time,sn-end-time,ip-protocol,ip-subscriber-ip-address,ip-server-ip-address,sn-subscriber-port,
sn-server-port,sn-nat-ip,sn-nat-port-block-start,sn-nat-port-block-end,sn-subscriber-nat-flow-ip,sn-subscriber-nat-flow-port,
sn-nat-realm-name,sn-nat-subscribers-per-ip-address,sn-nat-binding-timer,sn-nat-gmt-offset,sn-nat-port-chunk-alloc-dealloc-flag,
sn-nat-port-chunk-alloc-time-gmt,sn-nat-port-chunk-dealloc-time-gmt,sn-nat-no-port-packet-dropped,sn-closure-reason
,,,209.165.200.225,,,,209.165.201.1,1024,1039,,,NAT44_PUBLIC2,2,60,+0530,1,02/18/2020
06:41:08,,,
,,,209.165.200.225,,,,209.165.201.2,1024,1031,,,NAT44_PUBLIC5,8064,60,+0530,1,02/18/2020
06:41:14,,,
,,,209.165.200.225,,,,209.165.201.3,1024,1039,,,NAT44_PUBLIC2,2,60,+0530,0,02/18/2020
06:41:08,02/18/2020 06:42:12,,
,,,209.165.200.225,,,,209.165.201.14,1024,1031,,,NAT44_PUBLIC5,8064,60,+0530,0,02/18/2020
06:41:14,02/18/2020 06:44:24,,
```

Packet Drop EDR

Sample Packet Drop EDR

```
#sn-nat-no-port-packet-dropped,sn-start-time,sn-end-time,sn-subscriber-imsi
2,03/13/2020 08:28:24,03/13/2020 08:28:54,123456789012345
```



CHAPTER 54

NAT ALG Support

- Feature Summary and Revision History, on page 449
- Feature Description, on page 449
- Components of Session Initiation Protocol ALG, on page 450
- How it Works, on page 452
- NAT FW Processing, on page 454
- Configuring NAT ALG, on page 455
- Monitoring and Troubleshooting, on page 460

Feature Summary and Revision History

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Note This feature is not fully qualified in this release.

NAT performs translation service on any Transmission Control Protocol/User Datagram Protocol (TCP/UDP) traffic that doesn't carry source and/or destination IP addresses in application data stream. These protocols include:

- HTTP

- Trivial File Transfer Protocol (TFTP)
- Telnet
- Archie
- Finger
- Network Time Protocol (NTP)
- Network File System (NFS)
- Remote login (rlogin)
- Remote shell protocol (RSH)
- Remote copy protocol (RCP)

The following specific protocols have the IP address information within the payload. These protocols require the support of an Application Level Gateway (ALG) for translation services.

- FTP
- H323
- Session Initiation Protocol (SIP)
- Session Description Protocol (SDP)
- TFTP
- RTSP
- Point-to-Point Tunneling Protocol (PPTP)

Limitations

NAT64 to v4 translation for H323 is not supported.

Components of Session Initiation Protocol ALG

The following block diagram shows all the components that support SIP ALG for NAT or Firewall. The ALG-CORE and SIP APP are the new components. The other components are existing one which requires enhancements.

Note This example is specific to the SIP ALG, similar component is applicable for all other protocols in the document.

Figure 30: Components of Session Initiation Protocol (SIP) ALG

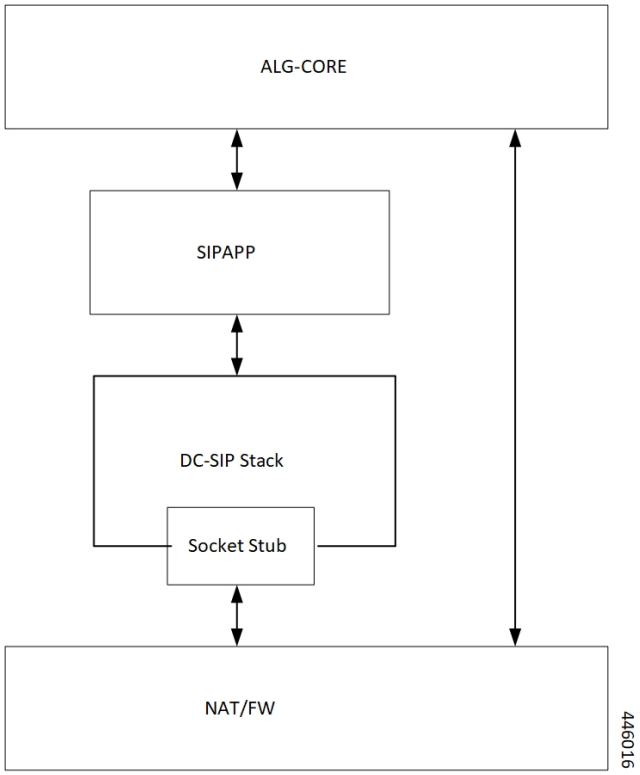


Table 30: Component and Functionality

Component	Function
ALG-CORE	• Interacts with the NAT/FW to create/modify/clear the pinholes. • ALG-CORE has the logic to store the pinhole information inside HA CLP. (defines a new pointer to structure called sip_alg_info). • ALG-CORE processes messages from SIPAPP based on the state and event it received.

Component	Function
SIP APP	• New functionality logic in each per request/response callback. • New data structures to maintain call/session related information's (based on stacks callCb/TransactionCb data structures). • Defines some generic UMM structures for sip/H.323, to interact with the ALG-CORE. • Do the encoding of the SIP message, that is private IP to public IP ...from the information returned by the ALG-CORE.
DC-SIP	DC-SIP is a full-blown sip stack, which parses the sip messages, maintains the transactions and call states. For SIP-ALG functionality the DC-SIP acts as B2BUA. Following are the functionalities DC-SIP stack provides: • Message parsing • Transaction management • Call management • Message encoding • Call back per request/response type.

The Socket Stub is the component that receives/sends packet from/to NAT/FW.

NAT/FW sends/receives the SIP packets to socket stub and it also provides the generic APIs to interact with ALG-CORE.

How it Works

Some of network applications exchanges the IP/Port information of server/client as part of payload. The server or client uses that exchanged IP/Port information to create new flows. As part of NAT ALGs, the server or client extracts that IP/Port info and allow those flows dynamically through pinholes.

In case of NAT, the server or client does the IP and transport level translations. The NAT IP and NAT Port replace the private source IP and source Port and conversely. But the sending application may not be aware of these translations since these translations are transparent.

For example, FTP NAT ALG function interprets the 'PORT' and 'PASV reply' messages. NAT translates the same in the payload so that the FTP happens transparently through the NAT.

NAT layer supports NAT 44 translation and NAT 64 translation. The NAT also supports 1:1 On demand NAT translation and Many:1 NAT translation.

Following are supported for each of the ALGs:

- NAT 44 1:1 On demand NAT translation
- NAT 44 Many:1 NAT translation
- NAT 64 1:1 On demand NAT translation
- NAT 64 Many:1 NAT translation

FTP

FTP is a TCP-based protocol and uses two flows one is for control messages another one for data/file transfer. FTP uses PORT and PASSIVE reply commands to exchange data flow parameters. These commands carry IP and Port information as part of the pay load.

RTSP

RTSP is a TC-based real time streaming protocol having different methods to control real-time media transfer. The control messages are having Port information embed, which is to transfer the media.

PPTP

Point-to-Point Tunneling Protocol (PPTP) allows the tunneling for Point to Point Protocol (PPP) through an IP network. PPTP uses an enhanced GRE (Generic Routing Encapsulation) to carry PPP packets. PPTP exchanges IP or port-specific information over its control connection and that information is to transfer the data over tunnel.

SIP

SIP is an application-layer control protocol. SIP can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. SIP is based on a request/response transaction model. Each transaction consists of a request that invokes a method, or function, on the server and at least one response. These requests and responses have client and server IP and port information. The SDP message bodies for describing multimedia sessions (that maybe present in SIP requests and responses) also has the IP and port information embedded in them.

For the transmission of media streams (voice, video) the SDP payload carried in SIP messages typically employs the Real-time Transport Protocol (RTP). The SIP ALG intercepts all the SIP communication and translates the private IP and port in the payload to NAT IP and Port.

TFTP

Trivial File Transfer Protocol (TFTP) is an application layer protocol for File Transfers. Due to its simple mechanism, many Embedded Systems uses this protocol to download images or files from the server. It’s a UDP-based protocol. TFTP L7 payload doesn’t contain IP or Port information but requires the pinholes to allow the Downlink initiated data flow.

H323

H323 is a set of protocol specifications that can establish, modify, and terminate multimedia sessions such as Internet telephony calls. Protocols involved in successful multimedia session are RAS, H225, H245, and media protocols (RTP, RTCP). RAS protocol is for communication between H323 Gatekeeper and the terminal. This communication helps to locate the other terminal to which it wants to communicate. H225 and H245 communicates between the terminals for session establishment, capability exchange, and media parameters exchange. The H245 messages have the details of the media channel in which the multimedia communication is going to take place. IP and Port information is present in the RAS, H225 and H245 messages. H323 ALG intercepts all the H323 communication and translates the private IP and port in the payload to NAT IP and Port.

NAT FW Processing

After receiving the key for processing the packets, the ECS framework creates flows with 5-tuple:

- Source IP
- Source port
- Protocol
- Destination IP
- Destination port

If it's the first packet with a given 5-tuple, then a NAT/FW rule match applies to check if the packet is acceptable or not. If packet is acceptable, then leads to a flow is creation.

Configuration of the NAT realm (NAT IP) is part of the rules. The NAT realm applicable for a flow is from the rule-definition that matches the packet

Rule configuration happens are based on well-known server addresses/port numbers. For example, the FTP service with port 21, SIP service with port 5060.

So, any FTP control session or SIP control session to well-known servers/port numbers finds a matching firewall rule. However, it may not be possible to configure rules for media flows (child flows) that are dynamically based on the control signaling.

In case of FTP data or SIP media packet, the NAT/FW rule definition match fails and drops the packets.

Another requirement is the control signaling and the corresponding media connection to use the same NAT realm. Same NAT IP address applies for control and media.

Even if the child flow (media connection) finds a matching NAT/FW rule. The child flow uses the NAT realm configuration for that rule, which isn't correct. The media flows should be using the same NAT realm that is applicable for the control connection.

So, the child flows even if there's no matching rule uses the same NAT realm that was for the control connection. In order to achieve the flow, create the pinholes based on the signaling messages. A pinhole contains subset of 5-tuple information.

Pinholes are to allow the traffic without doing any rule match (bypass rule match). The NAT realm is associated with the pinholes. Allows any traffic matching the pinholes and the NAT realm specified in the pinhole applies for noting the packets.

In case of many-to-one NAT, the NAT allows the downlink packets only if there's an active NAT binding. There are many services (SIP for example) where the remote end wants to initiate connections (incoming call). Under such conditions, to allow downlink packets the ALG needs to create required NAT bindings and associate with the pinholes by parsing signaling messages.

Following explains the uplink and downlink packet processing:

Uplink Packet Processing

Refer to the following points for the uplink packet processing.

- On receiving any uplink packet, comparison takes place against existing 5-tuple flows.
- If a matching flow exists (5-tuple match), the NAT binding that is associated with the flow applies on the packet.
- If no flow exists, then a pinhole lookup happens to check if there are any pinholes opened for this flow.
- If pinhole exists, then the NAT binding associated with the pinhole applies on the packet.
- If no pinhole exists, then rule match determines the NAT information for that flow. If no matching rule exists, the packet drops.

In case of outgoing SIP requests, the SIP message associates with the destination port as 5060. So, configure a rule with destination port as 5060 for identifying SIP traffic. The corresponding NAT realm configured for the rule gets applied on the SIP request.

Any pin holes based on the requests should have NAT bindings associated with them. This NAT bindings allocation is from the NAT realm that was for processing the request.

Downlink Packet Processing

Refer to the following points for the uplink packet processing.

- The downlink packets pass only if an active NAT binding exists. If the binding-look up fails, then the packet drops.
- If the binding lookup succeeds, the packet undergoes initial flow match processing same as an uplink packet processing.
- However, in case of downlink packets, no rule match happens for a packet from on a many-to-one NAT IP. The packet passes only if there's matching flow or a matching pinhole otherwise it drops. If a pinhole exists, then the NAT binding with the pinhole applies on that flow.
- In case of one-to-one NAT, even if there's no pinhole, rule match happens, and packet passes if a matching rule is there. The NAT realm that receives the packet applies for that downlink flow.

Configuring NAT ALG

Following are the commands to configure the NAT ALG.

```
configure
  active-charging service acs_service_name
```

```
    firewall nat-alg { default | no } { ftp | pptp | rtsp | sip | h323
}
    end
```

NOTES:

- **default**: Configures this command with the default setting for the specified parameter.
- **no**: Disables all/ or the specified NAT ALG configuration. When disabled, the ALG(s) will not do any payload translation for NAT calls.
- **ftp**: Enables/disables File Transfer Protocol (FTP) NAT ALG.
- **pptp**: Enables/disables Point-to-Point Tunneling Protocol (PPTP) NAT ALG.
- **rstp**: Enables/disables Real Time Streaming Protocol (RTSP) ALG.
- **sip**: Enables/disables Session Initiation Protocol (SIP) NAT ALG.
- **h323**: Enables/disables H323 NAT ALG.

Configuration for Many to One and One to Many

Many to one configuration on the User Plane.

```
ip pool NAT44_PUBLIC4 209.165.200.225 255.255.255.224 napt-users-per-ip-address 4 group-name
 NAT44_GRP2 on-demand max-chunks-per-user 4 port-chunk-size 32256
```

One to One configuration on the User Plane.

```
ip pool NAT44_PUBLIC4 209.165.200.225 255.255.255.224 nat-one-to-one on-demand group-name
NAT44_GRP1
```

Sample Configuration for FTP NAT ALG

In order to route the packets to the FTP ALG on Control Plane, Configure the following FTP routing rule.

```
Config
active-charging service acs
    ruledef rt_ftp-control
      tcp either-port = 21
      rule-application routing
      multi-line-or all-lines
    #exit
    ruledef rt_ftp-data
      tcp either-port = 20
      rule-application routing
      multi-line-or all-lines
    #exit
   access-ruledef SFW_HTTP
      ip any-match = TRUE
    #exit
    access-ruledef all
      ip any-match = TRUE
    #exit
    access-ruledef ipv6_nat
      ip server-ipv6-network-prefix = 64:ff98::/96
    #exit
    rulebase prepaid
      route priority 14 ruledef rt_ftp-data analyzer ftp-data
      route priority 15 ruledef rt_ftp-control analyzer ftp-control
```

```
    #exit
    fw-and-nat policy nat_policy1
      access-rule priority 1 access-ruledef ipv6_nat permit nat-realm NAT44_GRP1
      access-rule priority 10 access-ruledef SFW_HTTP permit nat-realm NAT44_GRP1
      access-rule priority 100 access-ruledef all permit nat-realm NAT44_GRP1
      nat policy ipv4-and-ipv6
    #exit
firewall nat-alg ftp ipv4-and-ipv6
  #exit
```

Sample Configuration for RTSP NAT ALG

Following are the sample configuration for RTSP NAT ALG:

```
Config
active-charging service acs
    ruledef rtsp-pkts
      tcp src-port = 554
      rule-application routing
    #exit
    ruledef rtsp-pkts1
      tcp dst-port = 554
      rule-application routing
    #exit
      access-ruledef SFW_HTTP
      ip any-match = TRUE
    #exit
      access-ruledef prefix1
      ip server-ipv6-network-prefix = 64:ff98::/96
    #exit
    rulebase cisco
     tcp 2msl-timeout 20
     tcp mss 1300 limit-if-present
      route priority 105 ruledef rtsp-pkts analyzer rtsp
      route priority 106 ruledef rtsp-pkts1 analyzer rtsp
      rtp dynamic-flow-detection
      fw-and-nat default-policy nat_policy1
#exit
fw-and-nat policy nat_policy1
      access-rule priority 1 access-ruledef prefix1 permit nat-realm NAT44_GRP1
      access-rule priority 10 access-ruledef SFW_HTTP permit nat-realm NAT44_GRP1
      nat policy ipv4-and-ipv6
    #exit
    firewall nat-alg rtsp ipv4-and-ipv6
```

Sample Configuration for PPTP NAT ALG

Following are the sample configuration for PPTP NAT ALG:

```
configure
active-charging service ACS
    ruledef pptp-route
      tcp either-port = 1723
      rule-application routing
      multi-line-or all-lines
     exit
     rulebase cisco
     route priority 1 ruledef pptp-route analyzer pptp
     #exit
#exit
access-ruledef all
```

```
      ip any-match = TRUE
#exit
access-ruledef ipv6_nat
ip server-ipv6-network-prefix = 101:101::/96
#exit
     rulebase cisco
     route priority 1 ruledef pptp-route analyzer pptp
     fw-and-nat default-policy nat_policy1
     #exit
      fw-and-nat policy nat_policy1
      access-rule priority 1 access-ruledef ipv6_nat permit nat-realm NAT44_GRP1
      access-rule priority 100 access-ruledef all permit nat-realm NAT44_GRP1
      nat policy ipv4-and-ipv6
    #exit
firewall nat-alg pptp ipv4-and-ipv6
#exit
```

Sample Configuration for TFTP NAT ALG

Following are the sample configuration for NAT44 on Control Plane:

```
configure
active-charging service ACS
    ruledef rt_tftp
      udp either-port = 69
      rule-application routing
      multi-line-or all-lines
    exit
      rulebase cisco
      route priority 1 ruledef rt_tftp analyzer tftp
   #exit
#exit
```

Following are the sample configuration for NAT64 on Control Plane:

```
conf
         active-charging service ACS
         ruledef rt_tftp
         udp either-port = 69
         rule-application routing
         multi-line-or all-lines
         exit
         access-ruledef all
         ip any-match = TRUE
         exit
         access-ruledef ipv6_nat
         ip server-ipv6-network-prefix = 64:ff98::/96
         exit
         rulebase cisco
         route priority 1 ruledef rt_tftp analyzer tftp
         fw-and-nat default-policy nat_policy
         #exit
        end
        conf
       context ISP1
       ip pool NAT44_PVT1 209.165.200.225 255.255.255.224 private 0 group-name NAT44_GRP1
       ip pool NAT44_PVT4 209.165.200.226 255.255.255.224 private 0 group-name NAT44_GRP1
      end
      conf
      context ISP1
      apn cisco.com
      ip address pool name NAT44_GRP1
      fw-and-nat policy nat_policy1
```

```
      exit
      end
     configure
     active-charging service ACS
     fw-and-nat policy nat_policy1
     access-rule priority 1 access-ruledef ipv6_nat permit nat-realm NAT44_GRP1
     access-rule priority 10 access-ruledef all permit nat-realm NAT44_GRP1
     nat policy ipv4-and-ipv6
     end
```

Sample Configuration for H323 NAT ALG

Following are the sample configuration for H323 NAT ALG:

```
configure
active-charging service ACS
     ruledef h323
      udp dst-port = 1719
      rule-application routing
    #exit
    ruledef h323_multi
      udp dst-port = 1718
      rule-application routing
    #exit
    ruledef h323_tcp
      tcp dst-port = 1720
      rule-application routing
    #exit
   rulebase cisco
   route priority 6 ruledef h323 analyzer h323
      route priority 7 ruledef h323_tcp analyzer h323
      route priority 8 ruledef h323_multi analyzer h323
      rtp dynamic-flow-detection
 fw-and-nat default-policy nat_policy1
     #exit
      fw-and-nat policy nat_policy1
      access-rule priority 100 access-ruledef all permit nat-realm NAT44_GRP1
      nat policy ipv4-and-ipv6
   #exit
firewall nat-alg h323 ipv4-only
#exit
```

Sample Configuration for SIP NAT ALG

Following are the sample configuration for SIP NAT ALG:

```
conf
active-charging service service_1
    ruledef sipalg
      udp dst-port = 5060
      rule-application routing
    #exit
    ruledef sipalg_tcp
      tcp dst-port = 5060
      rule-application routing
    #exit
   access-ruledef server2
      ip dst-address = 209.165.200.224/27
    #exit
    access-ruledef nat64
      ip server-ipv6-network-prefix = cccc:1111::/96
```

```
    ip any-match = TRUE
  #exit
  #exit
  rulebase base_1
    route priority 1 ruledef sipalg analyzer sip advanced description advanced
    route priority 2 ruledef sipalg_tcp analyzer sip advanced description advanced
    rtp dynamic-flow-detection
    fw-and-nat default-policy fw1
  #exit
  fw-and-nat policy fw1
    access-rule priority 2 access-ruledef server2 permit nat-realm natPool
    access-rule priority 3 access-ruledef nat64 permit nat-realm natPool
    nat policy ipv4-and-ipv6
  #exit
  firewall nat-alg sip ipv4-and-ipv6
  #exit
#exi
```

Monitoring and Troubleshooting

This section provides information on CLI commands that are available for monitoring and troubleshooting for NAT ALG feature in CUPS.

Show Commands and/or Outputs

This section provides information about show CLI commands that are available in support of NAT ALG feature in CUPS.

- **show user-plane-service statistics analyzer name rtsp**: Use this command to view RTSP-related statistics.

```
RTSP Session Stats:
   Total Uplink Bytes:             844   Total Downlink Bytes:          1440
   Total Uplink Pkts:               10   Total Downlink Pkts:              6
   Uplink RTP Bytes:                 8   Downlink RTP Bytes:         2851524
   Uplink RTP Pkts:                  2   Downlink RTP Pkts:             2741
   Uplink Retry Bytes:               0   Downlink Retry Bytes:             0
   Uplink Retry Pkts:                0   Downlink Retry Pkts:              0
   RTSP Sessions:                    1
```

- **show user-plane-service statistics analyzer name rtp**: Use this command to view RTP-related statistics.

```
RTP Session Stats:
   Total Uplink Bytes:               8   Total Downlink Bytes:       2851524
   Total Uplink Pkts:                2   Total Downlink Pkts:           2741

FastPath Statistics :
   Total FP Flows:                   1
   Total Uplink FP Bytes:            0   Total Downlink FP Bytes:    2850497
   Total Uplink FP Pkts:             0   Total Downlink FP Pkts:        2740
```

- **show user-plane-service statistics analyzer name rtcp**: Use this command to view RTCP-related statistics.

```
RTCP Session Stats:
      Total Uplink Bytes:          804   Total Downlink Bytes:           728
      Total Uplink Pkts:            16   Total Downlink Pkts:             13
```

- **show user-plane-service statistics analyzer name ftp**: Use this command to view FTP-related statistics.

```
FTP Session Stats:
   Current Control Sessions:         1         Current Data Sessions:         1
   Total Control Sessions:           1         Total Data Sessions:           3
   Uplink Control Bytes:           190         Downlink Control Bytes:      544
   Uplink Control Pkts:             23         Downlink Control Pkts:        15
   Uplink Data Bytes:             6733            Downlink Data Bytes:       12444
   Uplink Data Pkts:              5136         Downlink Data Pkts:           14
   Uplink Error Bytes:               0         Downlink Error Bytes:          0
   Uplink Error Pkts:                0         Downlink Error Pkts:           0
   Request Succeed:                 14         Request Failed:                0
   Unknown Requests:                 0         Unknown Responses:             0
   Uplink Bytes Retrans:             0         Downlink Bytes Retrans:        0
   Uplink Pkts Retrans:              0         Downlink Pkts Retrans:         0
   RETR commands:                    2         STOR commands:                 1
   Unknown packets received:                                                  0
   Data packet received without control connection:                           0
   Invalid packets:                                                           0
   Packets that could not be parsed:                                          0

FastPath Statistics :
   Total FP Control Flows:           0
   Total FP Data Flows:              3
 Uplink :                                  Downlink :
   Total FP Control Pkts :           0      Total FP Control Pkts :           0
   Total FP Control Bytes :          0      Total FP Control Bytes:           0
   Total FP Data Pkts :              0      Total FP Data Pkts :              0
   Total FP Data Bytes :             0      Total FP Data Bytes:              0
```

- **show user-plane-service statistics analyzer name pptp**: Use this command to view PPTP-related statistics.

```
PPTP Session Stats:
  Total Uplink Bytes:               0   Total Downlink Bytes:             0
  Total Uplink Pkts:                0   Total Downlink Pkts:              0
  Total GRE Sessions:               0   Invalid PPTP Pkts:                0
  Unknown PPTP Pkts:                0

  PPTP-GRE Session Stats:
  Total Uplink Bytes:               0   Total Downlink Bytes:             0
  Total Uplink Pkts:                0   Total Downlink Pkts:              0
```

- **show user-plane-service statistics analyzer name h323**: Use this command to view H323-related statistics.

```
H323 Session Stats:
  Total Uplink Bytes              0       Total Downlink Bytes             0
  Total Uplink Packets            0       Total Downlink Packets           0
  Total H323 calls                0
  Total RAS messages              0
  Total Q931 messages             0
  Total H245 messages             0
```

  - **show user-plane-service statistics analyzer name h323 protocol ras**: Use this command to view the h323 protocol ras statistics.

```
  Total RAS messages                  0
    RAS messages                             Uplink
  Downlink
    --------------                     ------------
  ------------
    GatekeeperRequest                          0
     0
    GatekeeperConfirm                          0
     0
    GatekeeperReject                           0
```

```
 0
RegistrationRequest                         0
 0
RegistrationConfirm                         0
 0
RegistrationReject                          0
 0
UnregistrationRequest                       0
 0
UnregistrationConfirm                       0
 0
UnregistrationReject                        0
 0
AdmissionRequest                            0
 0
AdmissionConfirm                            0
 0
AdmissionReject                             0
 0
LocationRequest                             0
 0
LocationConfirm                             0
 0
LocationReject                              0
 0
DisengageRequest                            0
 0
DisengageConfirm                            0
 0
DisengageReject                             0
 0
InfoRequest                                 0
 0
InfoRequestResponse                         0
 0
RequestInProgress                           0
 0
Unclassified                                0
 0
```

- **show user-plane-service statistics analyzer name h323**: Use this command to view H323-related statistics.

```
H323 Session Stats:
  Total Uplink Bytes              0       Total Downlink Bytes              0
  Total Uplink Packets            0       Total Downlink Packets            0
  Total H323 calls                0
  Total RAS messages              0
  Total Q931 messages             0
  Total H245 messages             0
```

  - **show user-plane-service statistics analyzer name h323 protocol h245** : Use this command to view the h323 protocol h245 statistics.

```
Total H245 messages                  0
 H245 messages                      Uplink                              Downlink

 -------------                     ------------
------------
 OpenLogicalChannel                          0
   0
 OpenLogicalChannelAck                       0
   0
 OpenLogicalChannelReject                    0
```

```
    0
  OpenLogicalChannelConfirm                  0
    0
  RequestChannelClose                        0
    0
  CloseLogicalChannel                        0
    0
  CloseLogicalChannelAck                     0
    0
  EndSessionCommand                          0
    0
  Unclassified                               0
    0
```

- **show user-plane-service statistics analyzer name h323 protocol q931** : Use this command to view the h323 protocol q931 statistics.

```
Total Q931 messages                  0
  Q931 messages                       Uplink                              Downlink

  -------------                      ------------
-------------
  Alerting                                   0
   0
  CallProceeding                             0
   0
  Setup                                      0
   0
  Connect                                    0
   0
  ReleaseComplete                            0
   0
  Facility                                   0
   0
  Progress                                   0
   0
  Information                                0
   0
  Unclassified                               0
   0
```

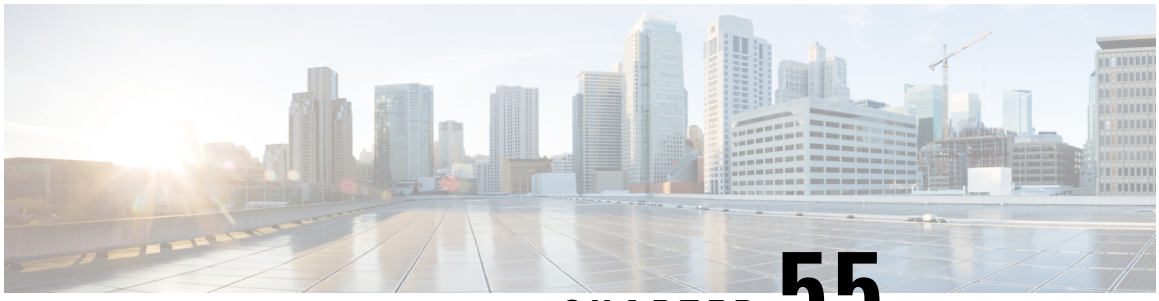
- **show user-plane-service statistics analyzer name tftp**: Use this command to view TFTP-related statistics.

```
TFTP Session Stats:
  Total Uplink Bytes:                0   Total Downlink Bytes:                0
  Total Uplink Packets:              0   Total Downlink Packets:              0
  Total Read Sessions:               0   Total Write Sessions:                0
  Total Invalid Control Packets:                                          0
  Total Invalid Data Packets:                                             0
  Total Packets with Unknown Request Type:                                0

  TFTP DATA Session Stats:
  Total Uplink Bytes:                0   Total Downlink Bytes:                0
  Total Uplink Packets:              0   Total Downlink Packets:              0
```

- **show user-plane-service statistics analyzer name sip**: Use this command to view SIP-related statistics.

```
SIP Session Stats:
      Total Uplink Bytes:              0          Total Downlink Bytes:          0
      Total Uplink Pkts:               0          Total Downlink Pkts:           0
      Uplink Valid Pkts:               0          Downlink Valid Pkts:           0
      Uplink Retry Pkts:               0          Downlink Retry Pkts:           0
      Uplink Error Pkts:               0          Downlink Error Pkts:           0
```

```
      Total SIP Calls:                  0
SIP Advanced Session Stats:
  Total Uplink Bytes                0   Total Downlink Bytes              0
  Total Uplink Packets              0   Total Downlink Packets            0

  Total SIP Calls                   0   Current SIP Calls                 0
  Total SIP UDP Calls               0   Current SIP UDP Calls             0
  Total SIP TCP Calls               0   Current SIP TCP Calls             0

  SIP Request                          Total received         Total transmitted
      Retransmitted
  -----------                    --------------------      --------------------
--------------------
  Register                                        0                        0
                0
  Invite                                            0                        0
                 0
  Ack                                               0                        0
                 0
  Bye                                               0                        0
                 0
  Info                                               0                        0
                  0
  Prack                                             0                        0
                 0
  Refer                                             0                        0
                 0
  Cancel                                           0                        0
                 0
  Update                                           0                        0
                 0
  Message                                         0                        0
                0
  Options                                          0                        0
                 0
  Publish                                           0                        0
                 0
  Subscribe                                       0                        0
                0
  Notify                                            0                        0
                 0


  SIP Response                         Total received         Total transmitted
      Retransmitted
  ------------                   --------------------      --------------------
--------------------
  1XX                                                0                        0
                  0
  2XX                                                0                        0
                  0
  3XX                                                0                        0
                  0
  4XX                                                0                        0
                  0
  5XX                                                0                        0
                  0
  6XX                                                0                        0
                  0
```



CHAPTER 55

N : M Redundancy

- Revision History, on page 465
- Feature Description, on page 465
- Configuring Ignore SSH IP Installation, on page 466

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
With this release support is provided for a new CLI to ignore the SSH IP installation in UP.	21.26.7
First introduced.	Pre 21.24

Feature Description

The CUPS User Plane (UP) is an all-important network component in the core network that carries and anchors the data traffic of subscribers. To ensure a smooth quality of experience (QoE), it is necessary to preserve data traffic and continue with minimal interruption. This is feasible only when there is a provision of a robust redundancy mechanism for all the data sessions that are hosted and anchored on UPs.

Every UP should have a redundant UP on standby (warm, hot, or active). However, this model mandates significant resource requirement for the service providers and is not a preferred model because of the number of UPs that can keep scaling horizontally. The preferred model is to have an N:M model with multiple UPs acting as standby-UPs for every active UPs. The N:M Redundancy feature provides this redundancy model.

On the UP, there is a new Cisco proprietary node called the Redundancy and Configuration Manager (RCM) which handles the configuration management of the UPs and the redundancy functionality.

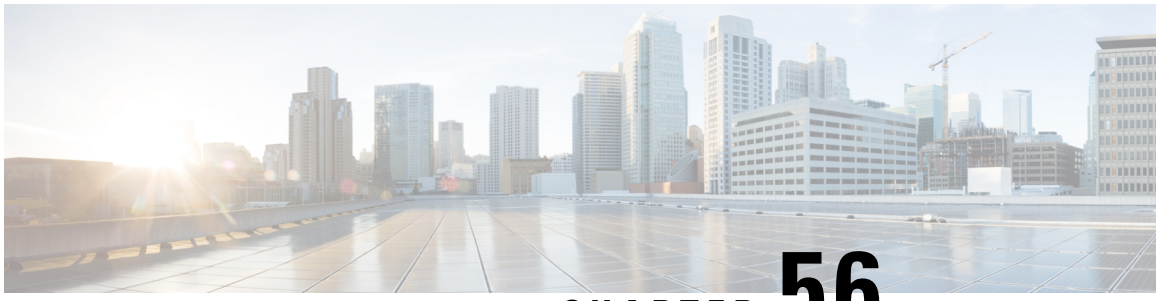
For details on N:M redundancy and RCM, refer the *Redundancy and Configuration Manager Configuration and Administration Guide*.

Configuring Ignore SSH IP Installation

Use the configuration given below to ignore the SSH IP installation in UP:

```
configure
   context context_name
      redundancy-configuration-module module_name
         ignore-ssh-ip
         end
```

Note By default, when this CLI is not configured, the NSO SSH IP is configured on UP as usual.



CHAPTER 56

Netloc and RAN/NAS Cause Code

- Revision History, on page 467
- Feature Description, on page 467
- Configuring Netloc and RAN/NAS Cause Code, on page 468

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The Netloc and RAN/NAS Cause Code feature is supported in non-CUPS architecture. With this release, this feature is qualified in CUPS architecture.

This feature is used to send detailed RAN and/or NAS release cause code information from the access network to PCRF.

This feature is in compliance with Release 12 specification of 3GPP TS 29.212.

If the supported features "netloc-ran-nas-code" and "netloc" are enabled, then netloc-ran-nas-cause code are sent to the PCRF through CCR-U/CCR-T message.

In the Charging-Rule-Report AVP and CCR-T, the Diameter AVP “RAN-NAS-Release-Cause” is included for bearer and session deletion events respectively when the NetLoc-RAN-NAS-Cause supported feature is enabled and the RAN/NAS cause is received from the access side.

In the CCR-U and CCR-T, the network location is sent in the Diameter AVP “3GPP-User-Location-Info” and/or “3GPP-MS-TimeZone” is included for creation/updation/deletion of bearer or session events respectively when the NetLoc-RAN-NAS-Cause supported feature is enabled and the Netloc is received from the access side.

Configuring Netloc and RAN/NAS Cause Code

Use the following configuration to enable the feature.

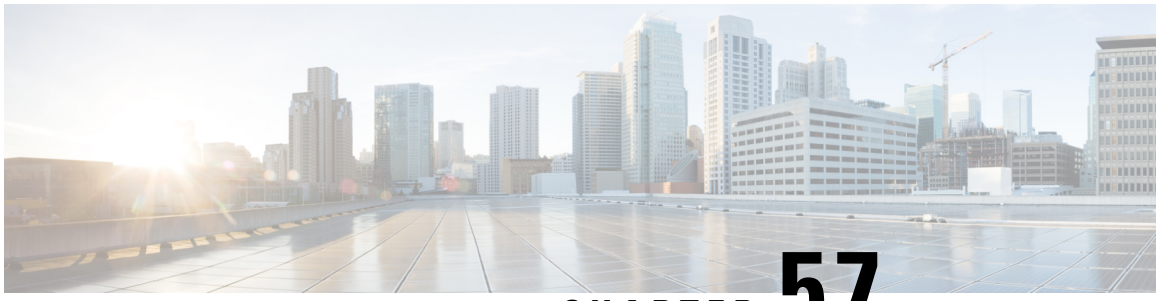
```
configure
   context context_name
      ims-auth-service service_name
         policy-control
            diameterencode-supported-featuresnetloc-ran-nas-cause
            end
```

NOTES:

- **netloc-ran-nas-cause**: Enables the Netloc-RAN-NAS-Cause feature. By default, this supported feature will be disabled.
- If the supported features "netloc-ran-nas-code" and "netloc" are enabled, then netloc-ran-nas-cause code will be sent to PCRF.
- To disable this supported feature, use the following command:

  [default | no] diameter encode-supported-features

- This feature is supported only for standard Gx dictionary (r8-gx-standard and dpca-custom8).



CHAPTER 57

Network Provided Location Indication

- Revision History, on page 469
- Feature Description, on page 469
- How It Works, on page 469

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

This feature enables the P-GW to provide the required access network information to the PCRF within the TWAN-Identifier AVP, User-Location-Info-Time AVP (if available), and/or UE-Local-IP-Address AVP as applicable for S2a/S2b. The P-GW also provides the ACCESS_NETWORK_INFO_REPORT event trigger within Event-Trigger AVP.

Important The Network Provided Location Indication (NPLI) is an existing feature that is supported in non-CUPS architecture. With this release, the feature is qualified in CUPS architecture. For more information, refer the *NetLoc for WiFi EPC* chapter in the *SAEGW Administration Guide*.

How It Works

During bearer deactivation or UE detach procedure, the P-GW provides the access network information to the PCRF within the TWAN-Identifier AVP and information on when the UE was last known to be in that location within User-Location-Info-Time AVP, and/or UE-Local-IP-Address AVP as applicable for S2a/S2b.

If the PCRF request for user-location information as part of the Required-Access-Info AVP and it is not available in the P-GW, then the P-GW provides the serving PLMN identifier within the 3GPP-SGSN-MCC-MNC AVP.

Previously, the P-GW notified ULI/MS-TimeZone/PLMN-ID to ECS/IMSA/PCRF only when their value changed. With this feature, the P-GW receives NetLoc indication in the rules sent by ECS regardless of whether the values changed, and it sends this to the ECS/IMSA/PCRF. If the P-GW receives NetLoc as "1", then it informs the MS-Timezone. If the P-GW receives NetLoc as '0', then it informs the ULI and ULI Timestamp. If ULI is not available in that case, then the PLMN-ID is sent. If NetLoc indication is received for an update, then the P-GW indicates this information to the access side in the UBReq using the RetLoc Indication flag.

This is required for VoLTE and aids in charging and LI functionality in IMS domain. This feature allows EPC to support an efficient way of reporting ULI and Time-Zone information of the subscriber to the IMS core network.

NOTE: In CUPS, when dedicated bearer is created by PCRF, it waits for CBRsp to trigger the CCR-I (for new bearer, NSAPI) towards OCS server. Since there is no usage for this bearer until this point, instead of sending a CCR-I with old access side information and following it up with a new CCR-U with updated access side information, the P-GW sends a single CCR-I message with updated access side information.

Supported Functionality

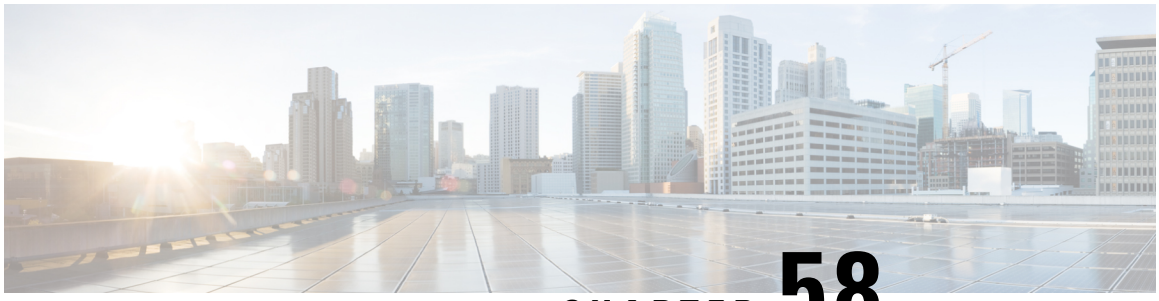
Netloc sent in CBRes/DSReq/UBRes/DBC/DBRes is supported on Gx, Gy, and Gz interfaces. The NPLI feature is supported for:

- Pure-P, Collapsed, and Pure-S sessions
- WiFi sessions
- S-GW Relocation
- Session Recovery

Limitations

The NPLI feature has the following limitations:

- GnGp handover scenarios are not supported.
- When there is a change in Netloc in UBRes, CDR for TimeZone change is not generated.
- When there is a ULI change in Netloc in DSReq, serviceConditionChange is blank in the CDR.



CHAPTER 58

Nexthop Forwarding Support IPv4/v6 Address

- Revision History, on page 471
- Feature Description, on page 471
- How It Works, on page 471
- Configuring Nexthop Forwarding Support IPv4/IPv6 Address, on page 475
- Monitoring and Troubleshooting, on page 476

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

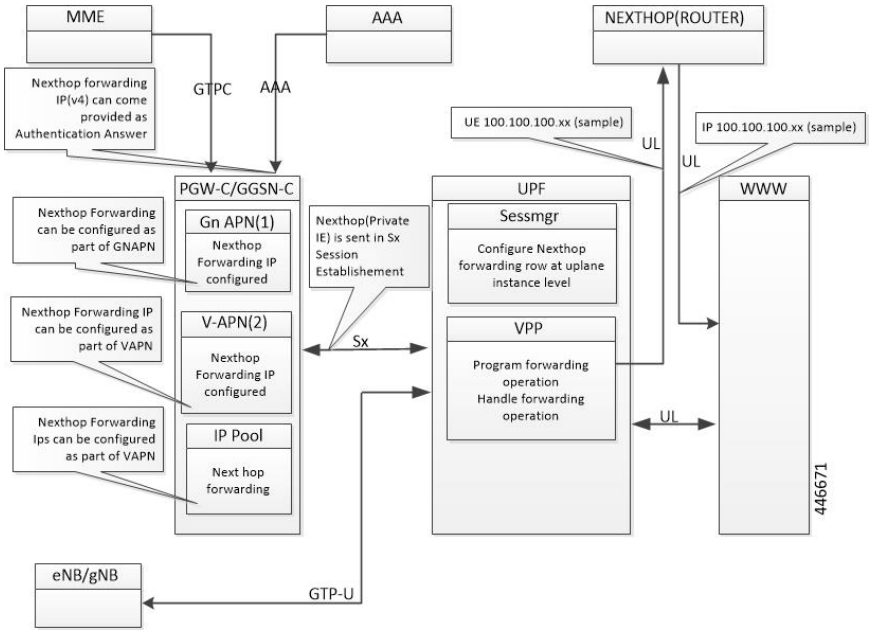
In uplink direction at CUPS UPF, UE IP and the GI IP might be in a different subnet and the routing path is defined to allow the uplink packet forward accordingly.

How It Works

Architecture

The following illustration provides EGCI-based P-GW UP Selection Solution overview.

Figure 31: Nexthop Forwarding



Configuration Priority

Configuration	Priority
AAA (Only IPv4)	1
APN (Gn/VAPN)	2
IP Pool	3

Configuration Use Cases

Case	IP Type	AAA	APN	IP Pool	Nexthop IP Selection
Nexthop supplied Only in AA message over AAA	IPv4	209.165.200.225	Not configured	Not configured	Nexthop Address Selected from AAA: IPv4: 209.165.200.225 IPv6: NA
	IPv6	Not supported	Not configured	Not configured	
Nexthop that is supplied in AA message over AAA + IPv4 configured in APN and IP Pool	IPv4	209.165.200.225	209.165.201.1	50.50.50.50	Nexthop Address is selected from AAA: IPv4: 209.165.200.225 IPv6: NA
	IPv46	Not supported	Not configured	Not configured	

Case	IP Type	AAA	APN	IP Pool	Nexthop IP Selection
IPv4 and IPv6 configured in APN only	IPv4	Not configured	209.165.201.1	Not configured	Nexthop Address is selected from APN: IPv4: 209.165.200.225 IPv6: 9001::3
	IPv6	Not supported	9001::3	Not configured	
IPv4 and IPv6 configured in IP Pool only	IPv4	Not configured	Not configured	50.50.50.50.	Nexthop Address is selected from IP Pool: IPv4: 209.165.200.225 IPv6 : 5002::5
	IPv6	Not configured	Not configured	5002::5	
IPv4 available over AAA + IPv4 and IPv6 configure on APN and IP Pool	IPv4	209.165.200.225	209.165.201.1	50.50.50.50	Nexthop IPv4 is selected from AAA: 209.165.200.225 Nexthop IPv6 selected from APN : 9001::3
	IPv6	Not Supported	9001::3	5002::5	
IPv4 available over AAA + IPv4 and IPv6 configure on IP Pool	IPv4	209.165.200.225	Not configured	50.50.50.50	Nexthop IPv4 is selected from AAA: 209.165.200.225 Nexthop IPv6 selected from IP Pool: 5002::5
	IPv6	Not Supported	Not configured	5002::5	
IPv4 available over AAA + IPv4 and IPv6 configure on APN.	IPv4	209.165.200.225	209.165.201.1	Not configured	Nexthop IPv4 is selected from AAA: 209.165.200.225 Nexthop IPv6 selected from APN: 9001::3
	IPv6	Not Supported	9001::3	Not configured	

Interface

Following Private IEs are introduced in SX Session Establishment message.

238	PFCP_IE_NEXTHOP	PFCP_IE_NEXTHOP								Sx Session Establishment Request	Private IE : CUPS: nexthop forwarding support-IPv4 /IPv6 address
		BITS									
		Octets	7	6	5	4	3	2	1		
		1 to 2	Type = 238 (decimal)								
		3 to 4	Length = n								
		5 to 10	PFCP_IE_NEXTHOP_ID								
		11-14	PFCP_IE_NEXTHOP_IP								

239	PFCP_IE_NEXTHOP_ID	PFCP_IE_NEXTHOP_ID								1. Inside create far IE of Sx Session Establishment Request	Private IE : CUPS: nexthop forwarding support-IPv4 /IPv6 address
		BITS								2. Inside PFCP_IE_NEXTHOP IE of Sx Session Establishment Request	
		Octets	7	6	5	4	3	2	1	PFCP_IE_NEXTHOP of Sx Session Establishment Request	
		1 to 2	Type = 239 (decimal)								
		3 to 4	Length = 5								
		5 to 10									

<table>
<tr><td rowspan="9">240</td><td rowspan="2">PFCP_IE_NEXTHOP_IP</td><td colspan="9">PFCP_IE_NEXTHOP_IP</td><td></td><td></td></tr>
<tr><td colspan="9">Bits</td><td rowspan="8">PFCP_IE_NEXTHOP of Sx Session Establishment Request</td><td rowspan="8">Private IE : CUPS: nexthop forwarding support-IPv4/ IPv6 address</td></tr>
<tr><td></td><td>Octets</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td></tr>
<tr><td></td><td>1 to 2</td><td colspan="7">Type = 240 (decimal)</td><td></td></tr>
<tr><td></td><td>3 to 4</td><td colspan="7">Length = n</td><td></td></tr>
<tr><td></td><td>5</td><td colspan="5">spare</td><td>V4</td><td>V6</td><td></td></tr>
<tr><td></td><td>m to m+3</td><td colspan="7">IPv4 Address</td><td></td></tr>
<tr><td></td><td>p to p+15</td><td colspan="7">IPv6 Address</td><td></td></tr>
</table>

Configuring Nexthop Forwarding Support IPv4/IPv6 Address

Configuring Nexthop Forwarding at APN Configuration Mode

Use the following CLI commands to configure Nexthop Forwarding at APN.

```
configure
   context context_name
      apn apn_name
         nexthop-forwarding-address { ipv4v6_address | ipv4_address | ipv6_address
}
         no nexthop-forwarding-address
         end
```

NOTES:

- **no**: Disables Nexthop forwarding address configuration.
- **nexthop-forwarding-address {** *ipv4v6_address* | *ipv4_address* | *ipv6_address* **}**: Configures the Nexthop forwarding address for this APN.
  - *ipv4_address* Configures IPv4 address.
  - *ipv6_address* Configures IPv6 address (supports colon-separated hexadecimal notation).

Configuring Nexthop Forwarding at IP Pool

Use the following CLI commands to configure Nexthop Forwarding at APN.

```
configure
   context context_name
```

```
[ no ] ip pool ipv4-public nexthop-forwarding-address  ipv4_address
[ no ] ip pool ipv6-public nexthop-forwarding-address  ipv6_address
end
```

NOTES:

- **no**: Disables Nexthop forwarding address configuration.
- **nexthop-forwarding-address** *ipv4_address* | *ipv6_address*: Configures the IPv4 address Nexthop forwarding address for this pool.
- **nexthop-forwarding-address** *ipv6_address*: Configures the IPv6 address Nexthop forwarding address for this pool.

Configuring Nexthop Forwarding Through AAA

Nexthop Forwarding Address can be configured through AAA. This option allows us to configure externally.

Configuring Nexthop Forwarding externally:

```
RADIUS AUTHENTICATION
  Access-Accept
   Subscriber-Nexthop-Address
```

Monitoring and Troubleshooting

This section provides information about CLI commands available for monitoring and troubleshooting the feature.

Show Commands and Outputs

This section provides information about show commands and their outputs in support of this feature.

show apn name <apn_name>

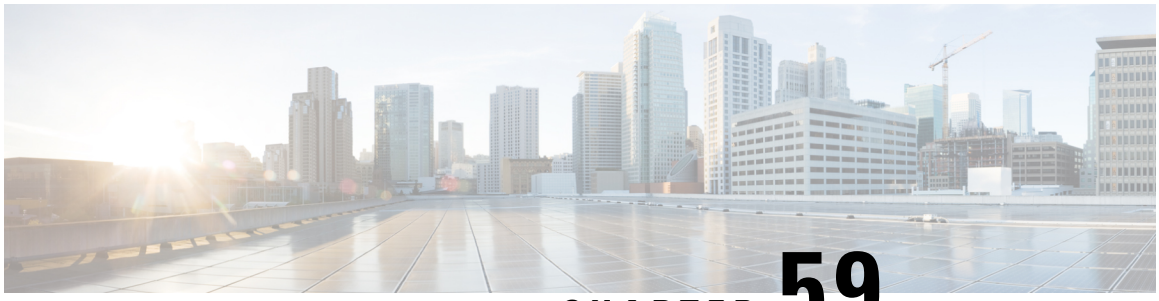
The output of this show command is enhanced to include the following fields introduced in support of this feature.

- nexthop gateway addr: Displays the configured Nexthop gateway address.

show subscriber user-plane-only full all

The output of this show command is enhanced to include the following fields introduced in support of this feature.

- Next Hop Ip Address - Displays the configured Nexthop IP address.



CHAPTER 59

Network Trigerred Service Restoration

- Feature Description, on page 477
- Configuring NTSR, on page 477
- Monitoring and Troubleshooting, on page 479

Feature Description

The Network Triggered Service Restoration (NTSR) feature detects an MME failure when enabled on the S-GW. If the subscriber served by the failed MME receives any downlink data packets, then the S-GW selects an alternate MME from the NTSR pool in round-robin fashion. The S-GW then sends a Downlink Data Notification (DDN) to the selected MME. This round robin selection of an MME is per session manager instance and not system wide.

The NTSR feature improves load balancing of DDN messages in the network during an MME failure.

In CUPS mode, bearers which are applicable for restoration, the corresponding downlink data is buffered on User Plane. For bearers that are not configured for restoration, the corresponding traffic endpoints are removed from the User Plane.

If S-GW detects that dedicated bearers are retained from a particular PDN, the S-GW retains the default bearer as well for this PDN. In this case, Downlink data will be dropped on default bearer.

On receiving any downlink data/Update Bearer Request/Create Bearer Request in restoration pending state, the SGW initiates a DDN request event towards MME or S4-SGSN.

Upon receiving Modify Bearer Request from MME, Control Plane sends Sx Session Modification Request to User Plane with UPDATE FAR:APPLY ACTION:FORW=1 for all bearers which are applicable for restoration.

Configuring NTSR

The NTSR feature involves the following configurations:

- APN Profile Configuration
- Peer Profile Configuration (Ingress)
- NTSR Pool Configuration
- S-GW Service Access Peer Map Association

- MME Restoration Timer Configuration

APN Profile Configuration

In this configuration, the QCI and ARP values are configured in the APN profile. When path failure is detected on the ingress side of the S-GW, bearers are retained or released based on the configured ARP/QCI values. S-GW can configure a maximum of two QCI and ARP-watermark combination per APN-profile.

Use the following commands to configure the ARP and QCI values in the APN profile.

```
configure
  apn-profile profile_name
    ntsr { all | qci qci_value | arp-priority-watermark arp_value }
    end
```

NOTES:

- **ntsr**: Specifies the NTSR configuration.
- **qci**: Specifies the QCI value for NTSR.
- **arp-priority-watermark**: Specifies the ARP value for NTSR.
- **all**: Identifies for all bearers with QCI or ARP values for MME restoration.

Peer Profile Configuration (Ingress)

In this configuration, the Peer Profile is configured on the ingress side of S-GW. The peer profile contains an associated pool-id, which is used to detect MME/S4-SGSN pool after MME failure.

Use the following commands to configure peer-profile on the ingress side at S-GW.

```
configure
  peer-profile service-type sgw-access name name
    ntsr pool-id pool_id
    end
```

NOTES:

- **sgw-access**: Configures the profile for peer nodes of S-GW towards S4/S11 interfaces.
- **ntsr**: Specifies the NTSR configuration.
- **pool-id**: Specifies the pool ID to detect MME/S4-SGSN pool after MME failure. The *pool_id* is an integer in the range of 1 to 10.

NTSR Pool Configuration

The NTSR pool configuration is used to configure pool of IP addresses associated with a pool-id and a peer type. One pool ID can be used for one peer-type. The NTSR pool can have combination of IPv4 or IPv6 address. S-GW can be configured with a maximum of 10 NTSR pools, and with at maximum of 5 IPv4v6 IP address pairs.

Use the following configuration to configure the NTSR Pool.

```
configure
  ntsr-pool pool-id pool_id peer-type [ mme | s4-sgsn ]
     [ no ] peer-ip-address { ipv4-address ipv4_address | ipv6-address
ipv6_address }
     end
```

NOTES:

- **pool-id**: Specifies the NTSR pool ID.
- **peer-type**: Specifies the NTSR Pool ID peer type. The peer type is either MME or S4-SGSN.
- **peer-ip-address**: Configures the IPv4 address or IPv6 address as a part of the MME or S4-SGSN pool.

S-GW Service Access Peer Map Association

In this configuration, the peer map on the Access side or Ingress side of S-GW service is configured.

Use the following configuration to associate a peer map to an S-GW service.

```
configure
  context context_name
    sgw-service service_name
      associate access-peer-map peermap_name
      end
```

NOTES:

- **access-peer-map**: Configures the Access/Ingress side peer map for an S-GW service.

Monitoring and Troubleshooting

Show Commands Input and/or Outputs

This section provides information regarding show commands and their outputs in support of the feature.

show apn-profile full all

The output of this command displays the following fields in support of this feature:

- NTSR
  - QCI
  - ARP-priority-watermark

show apn-profile full name *apn_name*

The output of this command displays the following fields in support of this feature:

- NTSR
  - QCI

- ARP-priority-watermark

show ntsr-pool all

The output of this command displays the following fields in support of this feature:

- SGW NTSR pools
- NTSR pool-id
- NTSR Pool type
- NTSR pool-id
- NTSR Pool type

show ntsr-pool full all

The output of this command displays the following fields in support of this feature:

- NTSR pool-id
- NTSR Pool type
- peer-address-pair(s)

show ntsr-pool full pool-id *pool_id*

The output of this command displays the following fields in support of this feature:

- NTSR pool-id
- NTSR Pool type
- peer-address-pair(s)

show ntsr-pool pool-id *pool_id*

The output of this command displays the following fields in support of this feature:

- NTSR pool-id
- NTSR Pool type

show sgw-service statistics all

The output of this command displays the following fields in support of this feature:

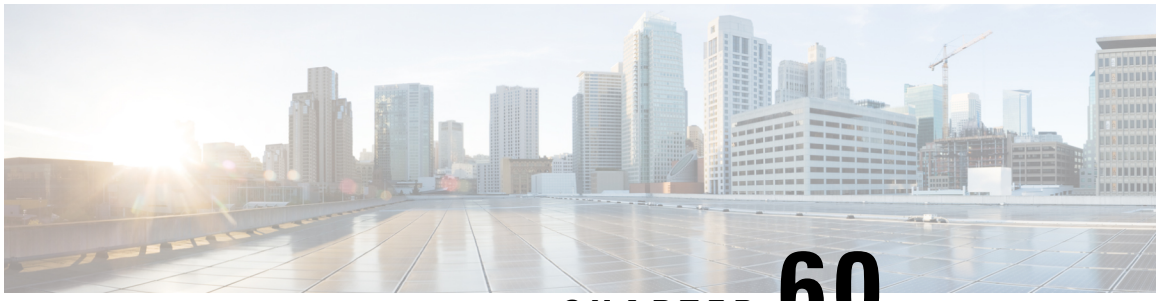
- Peer Failure
  - Retained
  - Restored
  - Released

- Peer Restart
  - Retained
  - Restored
  - Released

show subscribers sgw-only full all

The output of this command displays the following fields in support of this feature:

- NTSR state
- Bearer capable restoration



CHAPTER 60

NSO-based Configuration Management

- Revision History, on page 483
- Feature Description, on page 483
- How it Works, on page 484
- CUPS Configuration MOP, on page 494
- Troubleshooting , on page 523
- Appendix A: Incompatible StarOS Native Command Syntax, on page 524
- Appendix B: Example Configurations for N:M Deployment with RCM, on page 527

Revision History

Revision Details	Release
First introduced.	21.25

Feature Description

The Cisco Network Service Orchestrator (NSO) based configuration management for 4G CUPS supports:

- Onboarding of Cisco Virtual Network Function (VNF) devices—CP, UP, and RCM
- Centralized configuration management of 4G-based CPs, UPs, and RCMs for Day-N, Day-1, and Day-0.5 CUPS configuration push.
  - Day-0.5 applies to N:M UP redundancy scheme that uses RCM. The Day-0.5 configuration is intended for the UP to communicate to the RCM, so that it's role can be defined and suitable configuration be pushed subsequently.

Managing customer configuration management for 4G CUPS deployments using NSO automation also exhibits reusability, standard notification management, and systematic device configuration governance.

Use Cases

The NSO configuration-handling caters to the following use cases:

1. NSO on-boarding of VNFs (CPs, UPs, and RCMs) that are already deployed using Management IPs:
   - Onboarding of already-running VNFs (CPs, UPs, and RCMs) as devices into NSO and perform post-check to ensure the reachability and functioning of the devices. This is preliminary step to push/sync any configuration and establish communication for notifications.

     Orchestration of VNF devices (Instantiation and Destroy) is a separate module, and it doesn't have any dependency on configuration module. We need certain details (IP, Port, Management Username/Password) to onboard the device and supporting configuration management.

2. Allowing to store native-configs or device-templates for CPs, UPs, and RCMs:
   - Providing interface through RESTCONF/NSO-CLI to manage the reusable configurations for devices with logical name. Providing flexibility to network SMEs/Operators to create, modify, delete, and disable/enable the configurations. Aim is to pick those active configurations and apply to the device set as part of Day-0.5, Day-1, or Day-N for CPs, UPs, and RCMs.

3. Providing CLI/REST interface to apply Day-N/Day-1/Day-0.5 configurations to device logical groups (including CPs/UPs/RCMs) or custom list of target devices:
   - Providing interface through RESTCONF/NSO-CLI to Network SMEs/Operators to push Day-N/Day-1/Day-0.5 configurations to single or set of devices (CPs, UPs, or RCMs). This interface exhibits notifications/status on the progress of configuration push to the end users.

4. Logistic management of configuration management per device basis (Day-N, Day-1, or Day-0.5 pushed):
   - Providing dashboard utility and managing the configuration logs per device basis. This log is useful to know the most recent activity done on the device.

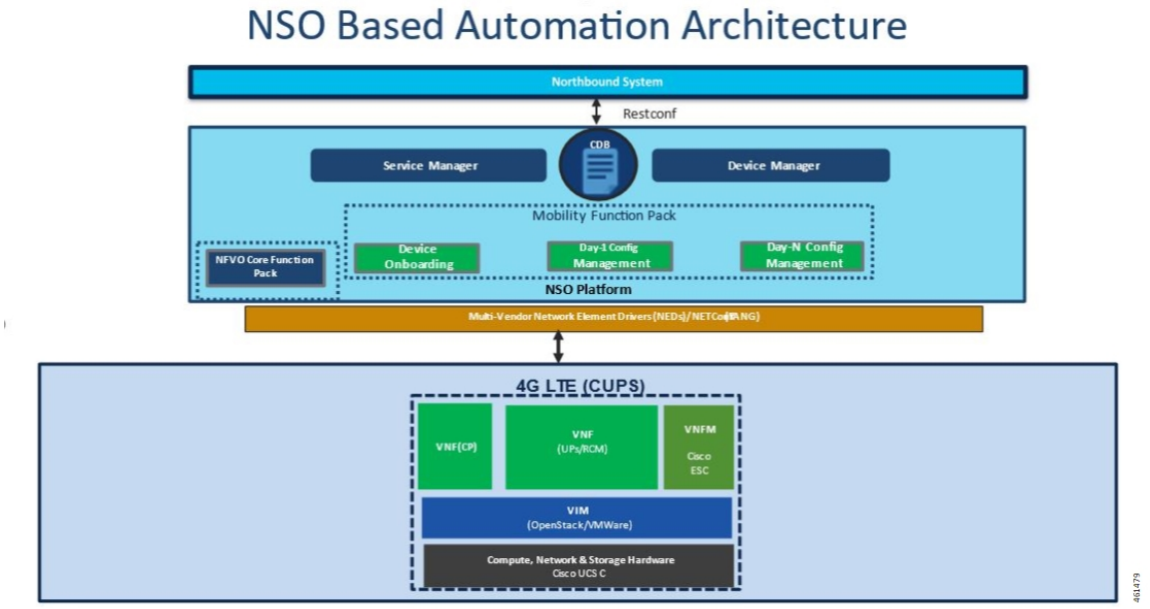
5. 5. Build the notification framework for RCM notification management in (N:M cases) and automate the configuration push for UPs for Day-N, Day-1, and Day-0.5:
   - Building notification framework in NSO to listen to RCM NetConf notifications on status changes, and push configurations automatically based on scenarios.

How it Works

-

Architecture

The following diagram illustrates, at a high-level, the components and frameworks involved in the solution.



RCM and NSO

In the N:M UP redundancy scenario, while the NSO manages configuration, the RCM will continue to arbitrate the role of the UPs (Active or Standby) and handle the switchover of an Active UP. Hence, this solution only moves the configuration function out of the RCM into the NSO; RCM is still required. For details about RCM, refer to the *RCM Configuration and Administration Guide*.

Components

Cisco 4G CUPS VNF deployment and configuration workflows are driven from the NSO. The following are some of the important components of NSO:

- **NSO Device Manager**: Manages each VNF component (CP, UP, RCM), keeps the copy of each device configurations, and manages the integrity of device configuration push.
- **NSO Service Manager**: Provides YANG standard to define high-level abstraction network service model for the customer/user input.
- **CDB**: Persistent Configuration Database for storing network configurations and operational data.
- **Mobility Function Packs**: Custom-built NSO packages to manage the 4G CUPS-based VNF orchestration and configuration management.
- **NFVO Core Function Pack**: NSO core NFV FP is a driver software to communicate with Cisco or other 3rd party VNFMs and VIMs, like OpenStack/VMWare, to deploy VNFs.
- **StarOS NSO NED**: StarOS-based NSO Network Element Driver (NED) that interfaces with the Cisco 4G CUPS VNFs for configuration push. This NED is based on Cisco CLI. The StarOS NSO NED communicates with the StarOS management CLI instance using Secure Shell (SSH).
- **RCM NSO NED**: RCM-based NETCONF NED is used to communicate with RCM devices for configuration management.

Minimum Platform, Hardware, and Software Requirements

The following are the minimum platform and software requirements to support centralized configuration management:

- Supported Orchestrator: NSO
- Configuration management for following Network Element's:
  - RCM: Redundancy and Configuration Management
  - VPC-SI: As 4G CUPS CP or UP
  - VPC-DI: As 4G CUPS CP only
- Minimum hardware requirements:
  - VM CPU: 8 CPU cores
  - VM RAM: 16 GB RAM baseline + 10 MB RAM for every StarOS device to be supported
  - VM connectivity: One 10 GBps network link. This can be used for both NSO HA and config/deployment by using separate VLANs or other mechanisms
  - VM Storage: 100 GB disk (preferably, SSD)
- Minimum software version

Software	Minimum Version
Cisco NSO	5.5.2
StarOS NSO NED	5.38.1
Cisco NSO HCC	5.0.0
Mobility Function Pack	3.0.0
StarOS version on UPs	21.25
StarOS version on CPs	21.25
RCM	21.25

Licensing

The NSO-based Configuration Management is a licensed Cisco feature. Contact your Cisco Account representative for detailed information on specific licensing requirements.

NSO Installation

-

Call Flows

This section describes the key call flows for the 4G CUPS configuration management functionality.

The call flows refer to NSO primitives like "connect", "sync-from", and so on. For detailed information on these primitives, refer to the *NSO User Guide*.

Note In the following call flow diagrams, “NSO Northbound” implies NCS CLI or RESTCONF interface.

Onboarding Existing 4G CUPS VNFs into NSO

This section describes the flow for adding existing 4G CUPS devices to NSO.

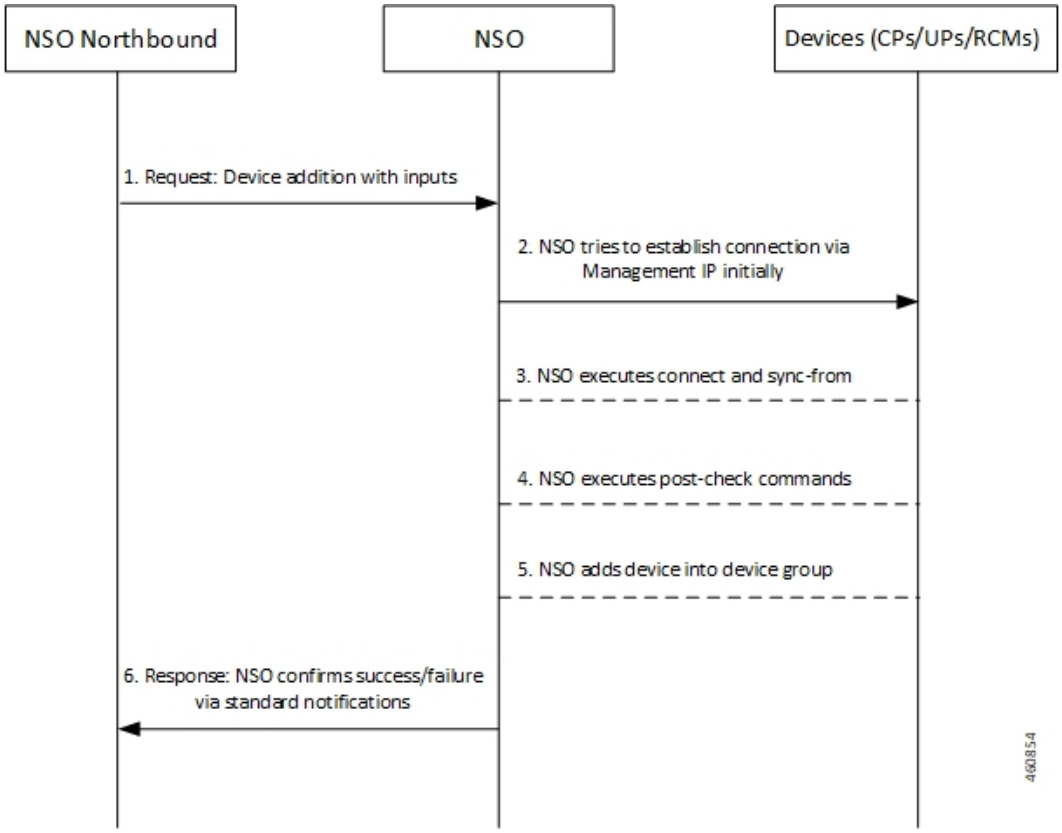


Table 31: Call Flow Description

Step	Description
1	NSO Northbound sends a request to NSO for adding devices into device-group. A device-group is a logical grouping of devices (VNFs) that share nearly identical configurations. This simplifies configuration in certain cases.
2	NSO initially attempts to establish a connection via Management IP.

Step	Description
3	NSO executes connect and sync-from commands. A sync-from operation pulls the existing configuration from the device/VNF into the NSO so that NSO is aware of the exact configuration on the device. The device configuration is not changed in a sync-from.
4	NSO executes post-check commands.
5	NSO adds device into device group.
6	NSO updates NSO Northbound about the success or failure of device addition via standard notifications.

4G CUPS Device Configuration Push – Manual

This section describes the flow for manual configuration push for 4G CUPS device.

This scenario applies to CPs, UPs, or RCMs in standalone or 1:1 redundancy configuration.

Prior to any configuration push, the device(s) must be onboarded. See Onboarding Existing 4G CUPS VNFs into NSO, on page 487.

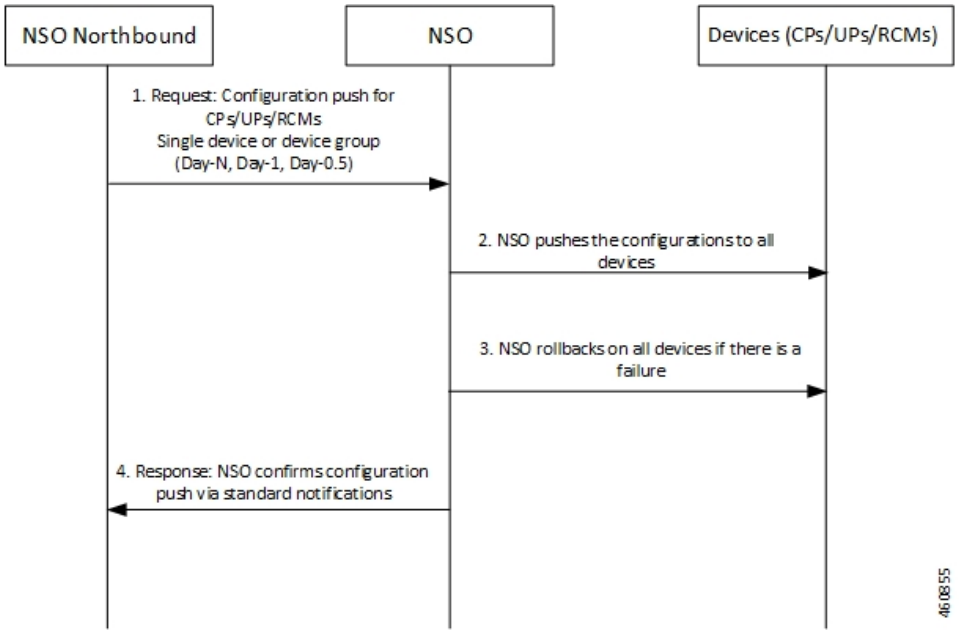


Table 32: Call Flow Description

Step	Description
1	NSO Northbound requests the NSO for a configuration push on devices such as CPs or Ups or RCMs. The devices can be a single device or a device group (Day-N, Day-1, Day-0.5).
2	NSO pushes the configurations to all devices.

Step	Description
3	If there is a failure, NSO rolls back the configuration on all devices. A rollback operation undoes the configuration applied to the device so as to restore it to the previous state (prior to application of the configuration).
4	NSO updates NSO Northbound about the configuration push via standard notifications.

Configuration Push from NSO to 4G CUPS UPs in N:M Redundancy – Automated

This section describes the flow for automatic configuration push from NSO to 4G CUPS devices.

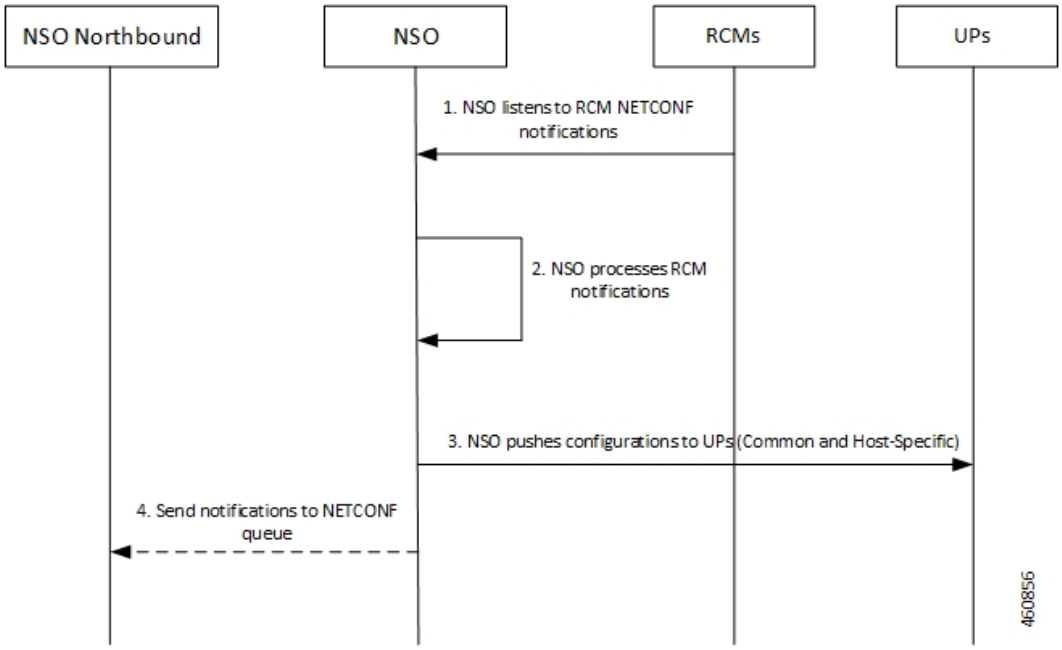


Table 33: Call Flow Description

Step	Description
1	NSO listens to RCM NETCONF notifications. When a UP connects to the RCM, the RCM decides what the UPs role should be (Active or Standby). This role is communicated in the notification. The configuration pushed to the UP is dependent on its role. So, the automated configuration push is driven based on the RCM notification.
2	NSO processes the received RCM notifications.

Step	Description
3	NSO pushes the common and host-specific configurations to UPs. Common configuration refers to the configuration that is shared across all the UPs in a redundancy group. This is typically Enhanced Charging Service (ECS) and Access Point Name (APN) configurations. Host-specific configuration is unique to an Active UP. Each Active UP needs its host-specific configuration. All Standby UPs need the host-specific configurations of all Active UPs, since Standby UP needs to be able to take over for any Active UP.
4	NSO sends notifications to NETCONF queue.

Configuration Metadata Pre-population

This section describes the flow for pre-population of configuration metadata

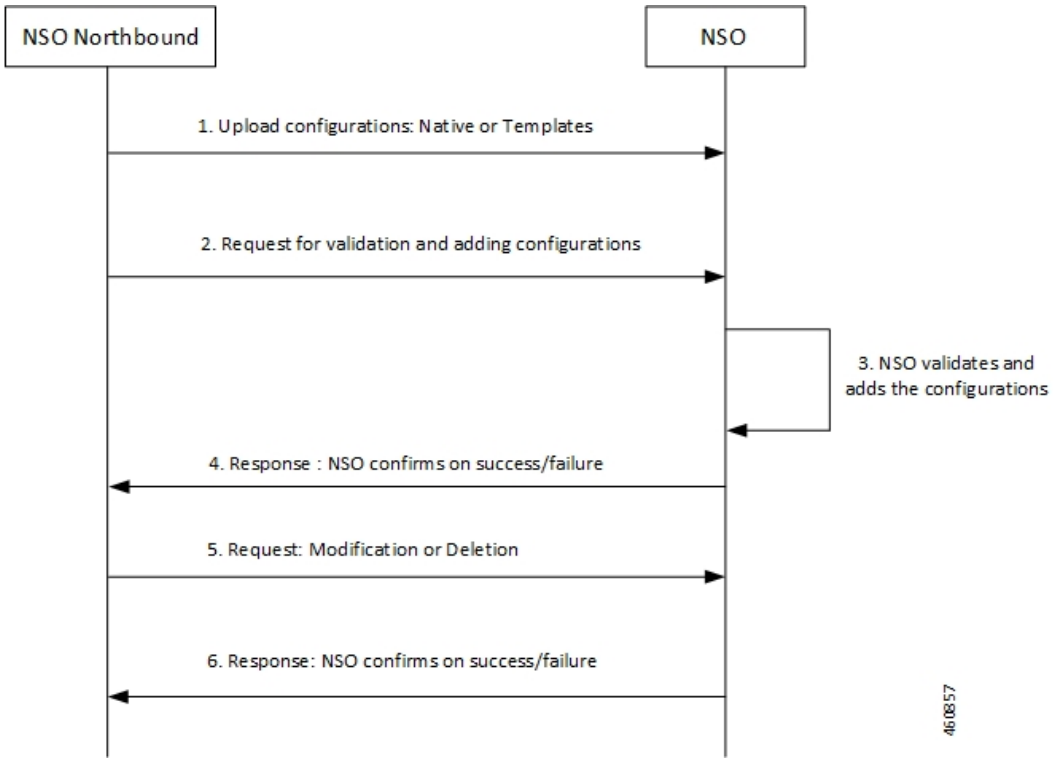


Table 34: Call Flow Description

Steps	Description
1	NSO Northbound uploads the configurations (native or templates) to NSO.
2	NSO Northbound requests NSO to validate and add configurations.
3	NSO validates and adds the configurations.
4	NSO updates NSO Northbound about the success or failure of device addition.

Steps	Description
5	NSO Northbound requests NSO for modification or deletion of configurations.
6	NSO updates NSO Northbound about the success or failure.

NSO HA Switchover Handling

This section describes the flow of handling NSO HA switchover.

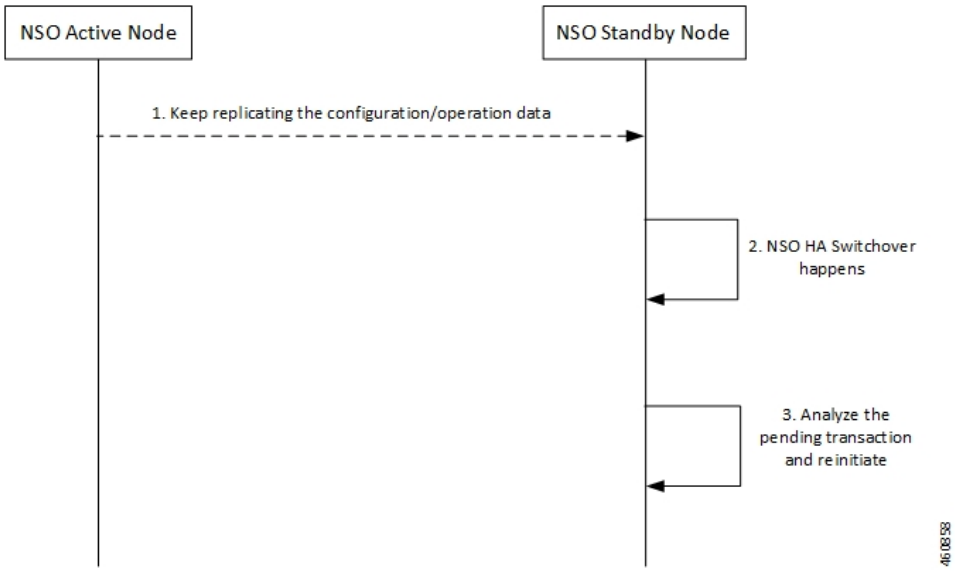


Table 35: Call Flow Description

Step	Description
1	NSO Standby Node keeps replicating the configuration or operation data from the NSO Active Node.
2	NSO HA switchover happens at the NSO Standby Node.
3	NSO Standby Node analyses the pending transaction and reinitiates the process.

Recovery

To recover from fault state to previous state, NSO provides in-built rollback mechanism for the pushed configurations. The following options are available for pushing the configuration to one or more devices:

- Commit or Dry-Runs only
- Commit with Rollback generated
- Scheme of Single or Multiple transactions
- Scheme for failure-handling on multiple transactions

- Scheme for pushing only stand-by nodes, active nodes, or common

CP Switchover (1:1)

The Mobility Function Pack does not actively track the active CP. It tracks on demand, if required, when a configuration push is initiated from northbound. Any configuration pushed to either CP is expected to be stored persistently as a boot configuration on that CP.

Note You must use the MOP option to save the configuration permanently.

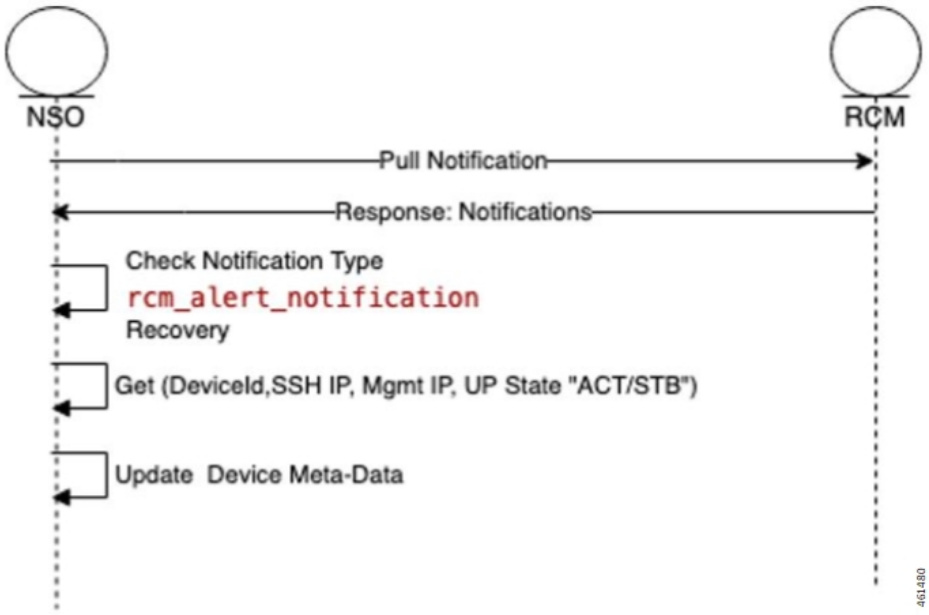
On a CP switchover, the CP that rebooted is expected to boot up with all the necessary configurations. The Mobility Function Pack does not perform any special handling in this scenario.

UP Switchover (1:1)

Like the CP scenario, any configuration pushed to either UP instance is saved persistently as part of the boot configuration. You must use the MOP option to save the configuration permanently. On a UP switchover, the UP that reboots are expected to come up with all the necessary configuration automatically. The Mobility Function Pack does not perform any special actions in this case.

UP Switchover (N:M)

The following figure illustrates the UP Recovery notification flow:



Once the NSO is subscribed to RCM device notification, NSO gets all notifications published to the stream “rcm-alert-notification”.

NSO performs the following steps whenever there is UP Recovery notification:

1. Waits for alert-status “Recovery”

2. Gets device details (Device Name, SSH IP, Management IP, UP Status)
3. Updates the device meta-data in NSO

Out-of-Band Configuration

A fundamental aspect of NSO-based configuration management is that the NSO maintains a copy of each device's (VNFs) configuration. When any configuration changes are applied from Northbound, the NSO compares its local copy of the configuration with the applied configuration to determine precisely what configuration needs to be pushed into the device. For this to occur, the NSOs copy of the configuration has to be in-sync with the actual configuration on the device/VNF.

There can be reasons due to which VNF configuration may be performed out-of-band, bypassing the NED. For example, any Day-0 configuration necessarily precedes onboarding the device into the NSO and thus is out-of-band (it is pushed through the appropriate VNF Manager). The configuration push MOP performs a "sync-from" operation prior to pushing any configuration to a device. This ensures that the NSO pulls any out-of-band configuration into the NSOs local copy, and the attempted configuration push is applied to the most current device configuration. The sync-from can only pull configuration that is known to the NED. Also, there are caveats when dealing with encrypted data.

Sensitive Elements in Configuration

StarOS encrypts sensitive elements in the configuration like passwords, keys, and so on. The encrypted items can only be decrypted by StarOS and are thus opaque to the NSO. Furthermore, the encrypted form of the sensitive item can change even when the underlying cleartext does not. As a result, the NSO cannot reliably detect any out-of-band changes made to such items.

The recommendation is to either:

- Completely manage the corresponding configuration out-of-band

—Or—

- Use only the NSO to manage the corresponding configuration, meaning the cleartext form of the command must be configured into NSO from Northbound initially, and for every subsequent change.

Do not mix NSO-based configuration management and out-of-band management for the same configuration.

Lawful Intercept

Lawful Intercept (LI) can be configured in a couple of different ways. One deployment involves all LI configuration in a single context (without using the dedicated LI context feature), and providing the general system administrator with LI administrator privileges. The other deployment involves a dedicated LI context, segregated LI configuration, and a dedicated LI administrator separate from general system administrator. There are other variations that likely fall in-between.

For the NSO to be able to manage LI configuration, it needs to:

- Have LI privileges and general system administrator privileges
- Be able to view and pull LI configuration in cleartext

For the deployment that involves all LI configuration in a single context scenario, the NSO must manage the LI configuration. For the other cases, it is recommended that the LI configuration be maintained out-of-band and provided as part of Day-0 configuration.

CUPS Configuration MOP

The Configuration MOP is the Method of Procedure (MOP) by which Configuration is applied to Cisco StarOS devices or RCM. This operation is invoked by network operator and in response, NSO provides the unique task-id for the request. Later, network operator can poll NSO using task-id to retrieve the status.

Configuration MOP broadly consists of the following three steps:

1. Device onboarding
2. Prepopulating Config metadata
3. Configuration push through Mobility MOP

Device Onboarding

The Device onboarding step is required only for the devices which are instantiated or orchestrated outside the Mobility orchestration solution. Otherwise, the instantiated VNF is implicitly onboarded onto the NSO as a device by the NSO-based Mobility orchestration solution.

Note This step is required only for the first time. Subsequent configuration pushes should skip this step.

The following examples illustrate how to onboard the VNF using RESTCONF or CLI respectively.

RESTCONF

Patch request for NSO URL: http://<NSO-IP>:<PORT>/restconf/data

The following is an example configuration:

```
{
  "data": {
    "nfv-device-onboarding:nfv-devices": {
      "device": [
        {
          "name": "<Device-or-VNF-Name>",
          "address": "<Management-Address>",
          "username": "<Management-Username>",
          "password": "<Management-Password>",
          "ned-type": "<cisco-staros/RCM>",
          "retry-options": {
            "number-of-attempts": <no-of-attempts-to-ping>,
            "delay": <delay-between-pings>
          }
        }
      ]
    }
  }
}
```

The following is an example configuration:

```
{
  "data": {
    "nfv-device-onboarding:nfv-devices": {
      "device": [
        {
          "name": "vpc-si25",
          "address": "209.165.200.225",
          "username": "admin",
          "password": "Cisco@123",
          "ned-type": "cisco-staros",
          "retry-options": {
            "number-of-attempts": 2,
            "delay": 10
          }
        }
      ]
    }
  }
}
```

CLI

You can also create/populate onboarding of the device using the following NSO CLI commands:

configure
nfv-devices device *device_name* **address** *ip_address* **username** *user_name* **password** *password* **ned-type cisco-staros retry-options delay** *delay_value* **number-of-attempts** *value*
commit

The following is an example configuration:

```
nfv-devices device dummy-device address 209.165.200.225 username admin password cisco@123
ned-type cisco-staros retry-options delay 10 number-of-attempts 2
```

Existing device in NSO can be deleted using the Delete request URL, no payload:
http://<NSO-IP>:<NSO-PORT>/restconf/data/nfv-device-onboarding:nfv-devices/device=<device-name>

It can also be deleted using NSO CLI:

configure
no nfv-devices device *device_name*
commit

Prepopulating Configuration Metadata

The configuration push MOP allows for variable substitution. This is useful in cases where nearly identical configuration is pushed to multiple devices (for example an ICSR active/standby pair). The differences can be represented as variables in the input configuration file. You can then populate the specific values for each device as metadata in a "variable: value" format. The MOP dynamically substitutes the right variable values at runtime.

If there are no prepopulated data for the device, Config MOP assumes that there are no dynamic substitution variables in config files, which are given for configuration push. If any attribute values that are referred in config files are missing, this step fails at runtime.

Prepopulating of config metadata has the following structure, and population of this data is based on the network scheme and data set. Highlighted items are mandatory for config push, and other items are optional.

```
container metadata-store{
    list config-metadata {
        key device-name;
        leaf device-name {
          tailf:info "onboarding device name";
          type string;
         }
 leaf schema {
   tailf:info "cluster-topology 1:1, N:M and N+2";
   type string;
 }
       list attributes {
         key attribute-name;
         leaf attribute-name {
           tailf:info "Attribute Name";
           type string;
         }
         leaf attribute-value {
           tailf:info "Attribute Value";
           type string;
           }
        }
 list configuration-type {
   key config-type;
   tailf:info "Configuration type Day0.5, Day1 or DayN";
   leaf config-type {
     type string;
    }
```

```
          list files {
      key file-name;
      tailf:info "file name";
      leaf file-name {
        type string;
      }
      leaf config-scheme {
        type string;
      }
   // CP device info
      list additional-files {
        key device; //cp device
        leaf device{
          tailf:info "device name";
          type string;
         }
    list additional-file{
      key additional-file-name;
      leaf additional-file-name{
        tailf:info "file name";
        type string;
        }
      }
        }
    }
  }
   }
}
```

Configuration meta-data is populated using the configuration meta-data request. This request follows the following YANG schema. The items in the "input" section are to be provided by the operator. The "output" section represents what is returned by the NSO after execution the action request.

The following is an example of NSO action to populate or modify the config meta-data:

```
tailf:action config-metadata-request {
          tailf:info "Invoke upgrade action on the selected devices";
          tailf:actionpoint config-metadata-request;
          input {
          list config-metadata {
   key device-name;
   leaf device-name {
     tailf:info "onboarding device name";
     type string;
   }

  list attributes {
  key attribute-name;
  leaf attribute-name {
           tailf:info "Attribute Name";
    type string;
  }
    leaf attribute-value {
    tailf:info "Attribute Value";
    type string;
        }
      }
    }
    }
    output {
  leaf status {
           type string;
         }
       }
   }
  }
```

RESTCONF

The following is an example to call this action from RESTCONF:

URI:
http://<NSO-IP>:<NSO-REST-PORT>/restconf/data/mobility-common:config-metadata/config-metadata-request

Content-Type: application/yang-data+json

Payload:

```
{
        "config-metadata": {
            "device-name": "test2",
     "schema" : "1:1",
            "attributes":{
                "attribute-name":"hostname",
                 "attribute-value": "TEST"
                  },
                "attribute-name":"BACKHAUL_IP",
                 "attribute-value": "209.165.200.225"
                }
        }
}
```

Result:

```
{
    "mobility-common:output": {
        "status": "Success"
    }
}
```

CLI

The following is an example to call this action using NCS CLI:

```
ubuntu@ncs> request config-metadata config-metadata-request config-metadata { device-name
staros-1 attributes { attribute-name hostname attribute-value TEST }
status Success
[ok][2021-07-12 08:05:01]
```

Configuration Push through Mobility MOP

This step is the final step in the configuration MOP. It allows you to push a fresh configuration or rollback a previously pushed configuration. The configuration to be pushed is present in one or more files as mentioned previously

Configuration MOP Push Request Flow

Network Operator invokes NSO API to start the process of config MOP automation for devices

NSO performs the below steps:

- Perform check-sync and sync-from or partial sync (if required) for the device. The check-sync determines if the NSOs copy of the device configuration is already in-sync with the actual device configuration.
- If specified in the MOP, NSO replaces the device attributes (variables) with node specific values read from device tree of the config metadata.
- NSO applies the configuration from the input files specified in the MOP to the device or list of devices in the order specified in the request. If there is a failure when pushing the configuration towards a device, then no further configuration is pushed to that device.
- NSO applies MOP(s) based on the mop type provided in the request (active/standby/common).
    - If the mop type is “common”, then NSO applies MOP(s) for all the devices provided in the request.

      In case of a failure, configuration push to the device(s) that encountered the failure is halted. Configuration push to other devices in the request continues. The Status shows the details of the failed devices. The operator then gets the option of rolling back the configuration on the failed device(s) as a separate action.
    - If the mop type is “active”, then NSO applies MOP(s) for all the “active” devices provided in the request

      The mop type “active” applies only to 1:1 redundancy scenario.

      In case of a failure, any pushed configuration is rolled back.
    - If the mop type is “standby”, then NSO applies MOP(s) for all the “standby” devices provided in the request.

      The mop type “standby” applies only to 1:1 redundancy scenario.

In case of a failure, any pushed configuration is rolled back.

- If the mop type is “pair”, then NSO applies MOP(s) first on the “standby” device and if successful, MOP(s) is applied on the “active” device. It performs the atomic transaction, so the configuration is applied to either both or neither device.

  The mop type “pair” applies only to 1:1 redundancy scenario.

  In case of a failure, any applied configuration is rolled back from both instances of the pair.

- If the mop type is “rcm-upf”, then NSO applies MOP(s) on the input device. Additionally, it identifies the RD-group of the input device and finds out the other UPF devices present in the same RD-group. Then it saves the ECS/APN config on the input device.

  In case of a failure, configuration push to the device(s) that encountered the failure is halted. Configuration push to other devices in the request continues. The Status shows the details of the failed devices. The operator then gets the option of rolling back the configuration on the failed device(s) as a separate action.

- NSO generates dry-run and reverse (rollback) configuration in native format for the MOP(s) supplied and stores in two separate files. In response, NSO returns both the file names along with absolute file path to the network operator.
  - Dry-run file is named as <MOP File Name>-<Device Name>-dryrun.txt.
  - Rollback file name is named as <MOP File Name>-<Device Name>-rollback.txt. Files are generated under the task id folder.

- If the network operator sends a request only for dry-run, then NSO generates dry-run and rollback files, but does not apply the MOP towards the device.
- If the network operator sends a request to apply the MOP, NSO generates dry-run and rollback files, and then applies the MOP towards the device.
- Network operator keeps on polling NSO for MOP automation status.
- NSO returns the list of hosts (devices) along with dry-run and rollback file location, and the status (Completed/In-Progress/Failed).

Configuration MOP Rollback Request Flow

- Network Operator invokes NSO API to start the process of rollback of a previously applied configuration.
- NSO performs the following steps:
  - Perform check-sync and sync-from or partial sync (if required) for the device.
  - NSO performs the rollback of MOP files in the reverse order for the task ID, MOP file name, and device name supplied by the network operator.

- If the MOP type is “pair”, then NSO performs rollback first on the “standby” device and after successful rollback, NSO performs rollback on the “active” device.
- If only task ID is supplied, then the whole transaction is rolled back. If task ID and MOP file name(s) are supplied, then only supplied MOPs are rolled back for all the devices. If task ID, MOPs file name, and devices names are supplied, then only supplied MOP for supplied devices are rolled back.

- NSO generates dry-run and reverse (rollback) configuration in native format for the rollback to be done, and stores in files. In response, NSO returns both file names along with absolute file path to the network operator.
  - Dry-run file is named as <MOP File Name>-<Device Name>-dryrun.txt
  - Rollback file is named as <MOP File Name>-<Device Name>-rollback.txt

  Files are generated under the task ID folder.
- If the network operator sends a request only for dry-run, then NSO generates dry-run and rollback files.
- If the network operator sends a request to roll back the MOP, NSO generates dry-run and rollback files, and then performs rollback.
- Network operator keeps on polling NSO for rollback status.
  - NSO returns the list of hosts (devices) along with dry-run and rollback file location, and the status (Completed/In-Progress/Failed)

MOP Automation

The Mobility configuration MOP is a set of commands that can be used to configure mobility devices from the NSO. This allows end user to specify locations to find or save the MOP-related input files and output files. It also allows the end user to setup global configurable parameters for MOP.

Configuration Prerequisites

- Navigate to NSO CLI and use static action to set the below parameters:
  1. Dry-run-mop location: Dry-run-mop file contains the configurations pushed to the device. Enter the location to save dry-run files of the MOP.
  2. Rollback-mop location: Rollback files are the configuration files generated that are required to roll back the configuration on device. Enter the location to save rollback files of the MOP.
  3. Config-mop-file location: Enter the location to fetch input configuration MOP files.
  4. Netconf-to-cli Conversion: Set the flag as “true” to convert NETCONF configuration to device CLI format. If the flag is set as “false”, then the dry-run file is generated in native NETCONF xml format.
- Static action call command in configuration:
- **static dry-run-mop /var/opt/ncs/**
  static rollback-mop /var/opt/ncs/

  To verify, use the following CLI command:

  show full-configuration static
- Global parameter configuration for mop-file location:

  ```
  configure
     configurable-parameters config-mop-file-loc /var/opt/ncs/
  ```

  To verify, use the following CLI command:

  show full-configuration configurable-parameters config-mop-file-loc

- StarOS-level NED setting examples

1. To prevent configuration update in the system cfg boot files of the devices, ensure that the write-memory-setting is disabled in the NCS CLI using the following command:

   ```
   devices global-settings ned-settings cisco-staros
   write-memory-setting disabled
   ```

2. Use the following command to exclude warnings while committing the configurations to the device:

   ```
   devices global-settings ned-settings cisco-staros behaviour
   config-warning-ignore.*Standby card not ready.*
   ```

Note Here, .*Standby card not ready.* can be replaced with the warning message to be ignored.

Mop-type Pair Prerequisites

- One of the device names, based on the identification of the device state (Active/Standby), can be specified as target-device-name.
- Configure the peer device and srp_loopback using the following commands:

```
config-metadata config-metadata-request config-metadata { device-name
up2-SI device-type vpc attributes { attribute-name srp_loopback
attribute-value 209.165.200.225 } scheme 1:1 }
config-metadata config-metadata-request config-metadata { device-name
up2-SI device-type vpc attributes { attribute-name Peer_Device_Name
attribute-value up1-SI } scheme 1:1 }
```

NSO APIs

NSO APIs are exposed by the NSO Mobility function pack that relate to configuration push functionality. These APIs are accessible either over RESTCONF or CLI.

Configuration Push MOP Automation

This API is used to start the MOP for pushing configuration to one or more devices. This is an asynchronous operation, and the status can be queried using a separate API.

API:

```
mop-automation
```

Request details

Parameter	Format	Required	Description
mop-file-name	List		
file-name	String	Key	Name of the device that corresponds to the UP/CP/RCM.
execution-order	Int	Mandatory	MOP execution order. 1 means first – order used to be 1,2,3 …

Parameter	Format	Required	Description
target-devices	List	Mandatory	Devices list
target-device-name	Leafref	Key	The NSO device name for the VNF. In case NSO is used for orchestration, the device name is the same as the VNF name.
operation-type	String	Mandatory	**`dry-run`** or **`commit`**

Parameter	Format	Required	Description
mop-type	Enum		Determines which device(s) to push the configuration to. Allowed values are: • **active** - Push to the active instance of a 1:1 redundancy pair. User inputs the device name for one of the instances of the redundancy pair (regardless of whether it is active or standby). The MOP automatically determines the currently active instance of that pair and pushes to that. • **standby** - Push to the standby instance of 1:1 redundancy pair. User inputs the device name one of the instances of the redundancy pair (regardless of whether it is active or standby). The MOP automatically determines the currently standby instance of that pair and pushes to that. • **pair** - Push to both active and standby instances of a 1:1 redundant pair. User inputs the device name for one of the instances of the redundancy pair (regardless of whether it is active or standby). The MOP automatically determines both the instances of the pair using the supplied instance, pushes to the standby instance first, and then to the active instance. • **common** - Push to all the devices provided in the request. This is the **default** . • **rcm-upf** - Push configuration to either single or all associated UPFs.

Parameter	Format	Required	Description
transaction-type	Enum		Allowed values are: • **single-transaction** - The configuration in all the supplied input files is combined and pushed to the device as a single transaction. • **multiple-transaction** - Each input file is pushed to the device as a separate transaction. This is the default. A transaction is an atomic unit of configuration change. All configuration in a transaction will either be pushed successfully or will be rolled back automatically if there was a failure during the push. **Note** A transaction will not span multiple devices. Each transaction is specific to a single device. So, for example, if the operator desires to push 3 files each to 2 devices, then: • with the multiple-transactions option, there will be a total of 3x2 = 6 transactions, one per device, per file. • with the single-transaction option, there will be a total of 2 transactions, 1 per device.

Response details

Parameter	Format	Required	Description
Task-id	String	Mandatory	Task unique identifier. The Task-id is to be used to query the status of the operation.
Time stamp	String	Mandatory	Time stamp
Error Code	String		Error Code
Error Message	String		Error Message

CLI

The following is an example of a request with NCS CLI:

```
mobility-mop:action mop-automation mop-type common transaction-type
multiple-transaction operation-type commit mop-file-name { file-name
dayN.txt order 1 target-devices-list { target-device-name up2-SI } }
```

REST API Request – Without Specifying Transaction Type

The following is an example of REST API request using postman without specifying the transaction type:

```
POST  /restconf/data/mobility-mop:action/mop-automation
   Host: localhost:8080
      Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache

         {
            “mop-automation”: {
            “operation-type”: “commit”,
            “mop-type”: “active”,
            “generate-dry-run”: “true”,
            “save-config-permanently”: “true”,
            “mop-file-name”: [
               {
                  “file-name”: “up_dayN.txt” ,
                  “order”: 1,
                  “target-devices-list”: [
                     {
                        “target-device-name”: “up2-SI”
                     }
                  ]
               }
          ]
     }
}
```

REST API Request – With Specifying Transaction Type

The following is an example of REST API request with specifying the transaction type:

```
POST    /restconf/data/mobility-mop:action/mop-automation
   Host: localhost:8080
      Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache
         Postman-Token: d2d2ddbe-5dff-4917-972a-146db6dc175f

            {
               "mop-automation": {
               "operation-type": "commit",
               "mop-type": "active",
               “transaction-type”: “single-transaction”,
               "mop-file-name": [
                  {
                     "file-name": "load3.txt",
                     "order": 1,
                     "target-devices-list": [
                        {
                           "target-device-name": "test3"
                        }
                     ]
                  },
                {
                     "file-name": "load4.txt",
                     "order": 2,
                     "target-devices-list": [
                 {
                     "target-device-name": "test3"
                 }
               ]
            }
         ]
```

```
    }
}
```

REST API Request – Without Specifying Transaction Type and mop-type as Pair

The following is an example of REST API request without specifying the transaction type and mop-type as pair:

```
 {
    "mop-automation": {
                          "operation-type": "commit",
                          "mop-type": "pair",
                          "generate-dry-run": "true",
                          "save-1-1-config": "true",
                          "mop-file-name": [
                    {
                       "file-name": "up_dayN.txt" ,
                       "order": 1,
                       "target-devices-list": [
                  {
                     "target-device-name": "up2-SI"
                  }
               ]
            }
         ]
    }
}
```

In response to successful invocation of above asynchronous requests, a unique task-id and time stamp is returned which is used to check the status of the mop-automation request.

```
{
    "mobility-mop:output": {
        "task-id": "1a1f62f0-487a-4c8c-bdeb-a760c26925cc",
        "time-stamp": "2021-07-19T11:10:51+0000",
        "time-zone": "Coordinated Universal Time"
    }
}
```

MOP Automation with mop-type as rcm-upf

Mop-type rcm-upf is used to push configuration to either single or all associated UPFs. The following two scenarios are applicable:

1. Apply MOP on single UPF.

   The following are ways to specify the UPF device:

   - Specify the upf-device in target-device-name.
   - Specify the rcm-vip, group, and device-id corresponding to upf-device.

   For the above two ways, the “only-to-target-devices” must set to “true” in the request.

   Payload Examples:

   a. Specify the upf-device in target-device-name:

   ```
   {
       "mop-automation": {
           "operation-type": "commit",
           "mop-type": "rcm-upf",
           "generate-dry-run": "true",
           "save-config-permanently": "true",
   ```

```
        "only-to-target-devices": "true",
        "mop-file-name": [
            {
                "file-name": "simpleStarOsChange.txt",
                "order": 1,
                "target-devices-list": [
                    {
                        "target-device-name": "up1-device"
                    }
                ]
            }
        ]
    }
}
```

b. Specify the rcm-vip, group, and device-id corresponding to upf-device:

```
{
    "mop-automation": {
        "operation-type": "commit",
        "mop-type": "rcm-upf",
        "generate-dry-run": "true",
        "save-config-permanently": "true",
        "only-to-target-devices": "true",
        "mop-file-name": [
            {
                "file-name": "simpleStarOsChange.txt",
                "order": 1,
                "rcm-vip" : "rcmvip01",
                "group" : "group03",
                "device-id" : "device-id1"
            }
        ]
    }
}
```

2. Apply MOP on all the associated UPF devices.

The following are the ways to identify the UPF devices:

- Specify a sample upf-device in target-device-name.
- Specify a rcm-vip and group.

Payload Examples:

a. Specify a sample upf-device in target-device-name:

```
  {
    "mop-automation": {
        "operation-type": "commit",
        "mop-type": "rcm-upf",
        "generate-dry-run": "true",
        "save-config-permanently": "true",
        "only-to-target-devices": "false",
        "mop-file-name": [
            {
                "file-name": "simpleStarOsChange.txt",
                "order": 1,
                "rcm-vip" : "rcmvip01",
                "group" : "group03"
            }
        ]
```

```
    }
}
```

b. Specify a rcm-vip and group.

```
{
    "mop-automation": {
        "operation-type": "commit",
        "mop-type": "rcm-upf",
        "generate-dry-run": "true",
        "save-config-permanently": "true",
        "only-to-target-devices": "false",
        "mop-file-name": [
            {
                "file-name": "simpleStarOsChange.txt",
                "order": 1,
                "rcm-vip" : "rcmvip01",
                "group" : "group03"
            }
        ]
    }
}
```

For the above two ways, the “only-to-target-devices” must be set to “false” in the request.

To retrieve the UPF device using rcm-vip and group, the data available in the following lists in CDB are used:

- device-id-up-mapping
- up-rcm-mapping
- rcm-upf-mapping

MOP Automation Status

NSO provides device status results for the task-id passed by the network operator.

API:

`mop-automation-status`

Request details

Parameter	Format	Required	Description
Task-id	String	Mandatory	Task unique identifier obtained from the "MOP Automation" API.

Response details

Parameter	Format	Required	Description
task-id	String	key	Task ID
task-status	String		Task status
Start-date	String		Start Date Time
End-date	String		End Date Time

Parameter	Format	Required	Description
Time-zone	String		Time zone
Operation-type	String		Commit/Dry-run
Action-type	String		Save
devices-list	list		Devices
Device-name	leafref	key	Device name
Start-date	String		Start Date Time
End-date	String		End Date Time
device-status	leafref		Device Status (Completed/In-Progress/Failed)
device-state	String		Active/Standby/Common /Pair/rcm-upf
files	list		Files
file-name	String	key	MOP file name
Order	Uint8		Order in which MOP has been performed
dry-run-mop	String		Dry-run output file location
rollback-mop	String		Rollback MOP location
Commit-queue-status	String		Commit queue status
Commit-queue-id	String		Commit queue ID
Error Code	String		Error Code
Error Message	String		Error Message
Error Code	String		Error Code
Error Message	String		Error Message

CLI

The following is an example of a request with NCS CLI:

```
mobility-mop:action mop-automation-status task-id
8d08e359-0bd2-48de-9a34-9192a986a486
```

REST API Request

The following is an example of REST API request to know the status of the mop-automation:

```
POST /restconf/data/mobility-mop:action/mop-automation-status
   Host: localhost:8080
      Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache

      {
         "task-id": "22301071-9a6c-4f27-a0dc-b50c24124806"
      }
```

The following is an example of response format generated, post invocation of the above request:

```
{
    "mobility-mop:output": {
        "task-id": "8d08e359-0bd2-48de-9a34-9192a986a486",
        "task-status": "COMPLETED",
        "start-date": "2021-09-06T09:08:54+0000",
        "end-date": "2021-08-06T09:09:10+0000",
        "time-zone": "Coordinated Universal Time",
        "operation-type": "commit",
        "action-type": "save",
        "devices-list": [
            {
                "device-name": "up2-SI",
                "device-status": "COMPLETED",
                "start-date": "2021-08-06T09:08:55+0000",
                "end-date": "2021-08-06T09:08:59+0000",
                "device-state": "active",
                "files": [
                    {
                        "file-name": "up_dayN.txt",
                        "order": "1",
                        "dry-run-mop":
"/var/opt/ncs//8d08e359-0bd2-48de-9a34-9192a986a486/up2-SI/up_dayN_commit_2021-08-06T090854+0000.txt",

                        "rollback-mop":
"/var/opt/ncs//8d08e359-0bd2-48de-9a34-9192a986a486/up2-SI/up_dayN_rollback_commit_2021-08-06T090854+0000.txt",

                        "commit-queue-status": "completed",
                        "commit-queue-id": "1628240937590"
                    }
                ]
            }
        ]
    }
}
```

MOP Rollback

NSO starts the process of rollback for task ID, MOP file, and devices provided in the input of the request.

This is the only option to roll back the MOP-configured configs or rolled-back configs.

This API rolls back a previously applied configuration. This uses the rollback files creating while originally applying the configuration. Rollback can be done per file, or for all files, and per device, or for all devices.

Note The success of a rollback is highly dependent on what changes have been made to the system since the relevant configuration was pushed. Subsequent changes may have changed the system such that the rollback configuration may not make sense.

API:

`mop-rollback`

Request details

Parameter	Format	Required	Description
Task-id	String	Mandatory	Task Unique Identifier

Parameter	Format	Required	Description
mop-file-name	List	Optional	MOP files along with device
file-name	String	Key	The original file name, used for identifying the corresponding rollback file.
target-devices	List	Optional	Device list. If not provided, the rollback is performed on all devices to which configuration was pushed in the original transaction.
target-device-name	Leafref	Key	
operation-type	String	Mandatory	Dry-run/Commit

Response details

Parameter	Format	Required	Description
Task-id	String	Mandatory	Task unique identifier. The task-id is to be used to check the status of the rollback via a separate API.
Time stamp	String	Mandatory	Time stamp
Error Code	String	Error Code	
Error Message	String	Error Message	

CLI

The following is an example of a request with NCS CLI:

```
mobility-mop:action mop-rollback task-id
8d08e359-0bd2-48de-9a34-9192a986a486 generate-dry-run true operation-type
 commit save-config-permanently true mop-file-name { file-name up_dayN.txt
 target-devices-list { target-device-name up2-SI } }
```

REST API Request – With Operation Type "commit"

The following is an example of REST API request with operation-type "commit":

```
POST /restconf/data/mobility-mop:action/mop-rollback
   Host: localhost:8080
      Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache
         Postman-Token: 1b687031-dc32-41 14-a69f-5984130c36a5
         {
            "mop-rollback": {
            "task-id": "0891655c-642b-4ba3-9392-6f05d4e77a63",
            "operation-type": "commit",
```

```
            "generate-dry-run": "true",
            "save-config-permanently": "true",
            "mop-file-name": [
            {
               "file-name": "up_dayN.txt",
               "target-devices-list": [
               {
                  "target-device-name": "up2-SI"
               }
            ]
         }
      ]
   }
}
```

In response to a successful invocation of above request, a unique task-id and time stamp is returned which is used to check the status of the mop-rollback request.

```
{
    "mobility-mop:output": {
        "task-id": "8d08e359-0bd2-48de-9a34-9192a986a486",
        "time-stamp": "2021-08-06T09:08:44+0000",
        "time-zone": "Coordinated Universal Time"
    }
}
```

REST API Request – With Operation Type “dry-run”

The following is an example of REST API request with operation-type "dry-run":

```
POST /restconf/data/mobility-mop:action/mop-rollback
   Host: localhost:8080
      Authorization: Basic YWRtaW46YWRtaW4= Content-Type:
application/vnd.yang.data+json cache-control: no-cache
      Postman-Token: 1b687031-dc32-41 14-a69f-5984130c36a5
      {
         "mop-rollback": {
         "task-id": "0891655c-642b-4ba3-9392-6f05d4e77a63",
         "operation-type": "dry-run",
         "generate-dry-run": "true",
         "save-config-permanently": "true",
         "mop-file-name": [
         {
            "file-name": "up_dayN.txt",
            "target-devices-list": [
            {
                  "target-device-name": "up2-SI"
               }
            ]
         }
      ]
   }
}
```

In response to a successful invocation of above request, a unique task-id and time stamp is returned which is used to check the status of the mop-rollback request.

```
{
    "mobility-mop:output": {
        "task-id": "1a1f62f0-487a-4c8c-bdeb-a760c26925cc",
        "time-stamp": "2021-07-19T11:10:51+0000",
        "time-zone": "Coordinated Universal Time"
}
```

MOP Rollback Status

NSO provides device status results for the task-id passed by the network operator. API to query the status of an ongoing or completed rollback operation.

API:

`mop-rollback-status`

Request details

Parameter	Format	Required	Description
Task-id	String	Mandatory	Task unique identifier of the rollback operation.

Response details

Parameter	Format	Required	Description
task-id	String	key	Task unique identifier
task-status	String		Task status
Start-date	String		Start Date Time
End-date	String		End Date Time
Time-zone	String		Time zone
Operation-type	String		Commit/Dry-run
Action-type	String		Rollback
devices-list	list		Devices
Device-name	leafref	key	Device name
Start-date	String		Start Date Time
End-date	String		End Date Time
device-status	leafref		Device Status (Completed/In-Progress/Failed)
device-state	String		Active/Standby/Common/Pair
files	list		Files
file-name	String	key	MOP file name
Order	Uint8		Order in which MOP has been performed
dry-run-mop	String		Dry run output file location
rollback-mop	String		Rollback MOP location
Commit-queue-status	String		Commit queue status
Commit-queue-id	String		Commit queue ID
Error Code	String		Error Code
Error Message	String		Error Message

Parameter	Format	Required	Description
Error Code	String		Error Code
Error Message	String		Error Message

CLI

The following is an example of a request with NCS CLI:

```
mobility-mop:action mop-rollback-status task-id
fd0fb9ae-8685-420e-9490-0c6858d14148
```

REST API Request

The following is an example of REST API request to know the status of the mop-rollback:

```
POST /restconf/data/mobility-mop:action/mop-rollback-status
Host: localhost:8080
Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache
Postman-Token: Oe2c4bd3-2dc6-4ddb-aea9-1 1 Occf622da7
"mop-rollback -status": {
"task-id": "5733d661-9242-4867-8320-a314da592c93"
}

Below is response format generated, post invocation of the above request -
task-id fd0fb9ae-8685-420e-9490-0c6858d14148
task-status COMPLETED
start-date 2021-08-06T09:24:14+0000
end-date 2021-08-06T09:24:30+0000
time-zone Coordinated Universal Time
operation-type commit
action-type rollback
devices-list {
    device-name up2-SI
    device-status COMPLETED
    start-date 2021-08-06T09:24:14+0000
    end-date 2021-08-06T09:24:19+0000
    device-state active
    files {
        file-name up_dayN_rollback_commit_2021-08-06T090854+0000.txt
        order 1
        dry-run-mop /var/opt/ncs//fd0fb9ae-8685-420e-9490-0c6858d14148/up2-SI
/up_dayN_2021-08-06T090854+0000_rollback_commit_2021-08-06T092414+0000.txt
        rollback-mop /var/opt/ncs//fd0fb9ae-8685-420e-9490-0c6858d14148/up2-SI
/up_dayN_2021-08-06T090854+0000_commit_2021-08-06T092414+0000.txt
        commit-queue-status completed
        commit-queue-id 1628241856973
    }
}
```

Verifying the dry-run and Reverse dry-run MOP

To verify the dry-run MOP and reverse dry-run MOP, go to respective file location which was provided while configuring static data for dry-run MOP and reverse dry-run MOP.

Adding Variables to Configuration File for MOP Execution

The MOP automation package supports specifying variables in the MOP so that they are populated at runtime based on what device the MOP is being applied to. For example, if the following MOP was specified and was executed on device TXPCF003:

```
config context local administrator $Host_name password Nsotest123$ exit
end
```

The Host Name can be configured using the following action call:

```
config-metadata config-metadata-request config-metadata { device-name
up2-SI device-type vpc attributes { attribute-name Host_name
attribute-value TXPCF003} scheme 1:1 }
```

The dry-run MOP that would be generated is as follows:

```
config context local administrator TXPCF003 password Nsotest123$ exit end
```

UP Configuration Push and Recovery in N:M Redundancy

In the N:M scenario, the RCM determines the role of each UP (active versus standby). Since any of the M standby instances must be capable of taking over for any of the N active instances, the configurations to be pushed are different and dynamic. This also means not all configuration can be saved on the UPs persistently.

RCM issues NETCONF notifications for relevant events such as a UP booting up or UP switchover. NSO listens to those notifications and applies the necessary configuration as appropriate.

The configuration for a UP consists of the following logical components:

- Day-0 configuration: This is primarily basic configuration for the UP's management interface to be reachable. This is pushed at the time of UP deployment by the VNFM. This configuration is expected to be persistent across reboot.

Note The require

```
rcm-configmgr CLI
```

command must be configured on the UP as part of Day-0 configuration for the NSO-based configuration push to work. This command is required irrespective of whether RCM is used in the solution or not. Without configuring this command, the ECS configuration push appears hidden.

- Day-0.5 configuration: This is configuration that allows the UP to contact the RCM. This configuration can be pushed either along with day 0, or pushed separately from the NSO, either automatically, right after UP deployment (if NSO is deploying the VM), or by a manual execution of the config push MOP. This configuration is also expected to be saved persistently across reboots.
- Common configuration: This is configuration that is common to all UPs regardless of whether they are active or standby. This is ECS and APN configuration only. This configuration needs to be pre-populated in the NSO. NSO will push this upon receiving notifications from the RCM. This configuration is not saved persistently as part of the boot configuration but is saved locally as a file on each UP and re-applied on every reboot by the NSO automatically.

- Host-specific configuration: This is configuration that is unique to each active UP. This is primarily the various service IP addresses. Each active UP is pushed the configuration specific to that active instance. Each standby instance is pushed the combined host-specific configurations of all active UPs. This configuration is expected to be pre-populate don the NSO. NSO will push this to each UP as appropriate based on the notification from the RCM.
- Host-specific configuration - RCM copy: This is the host-specific configuration of each UP, however, formatted in RCM compatible format. This needs to be pushed to the RCM. While RCM is not involved in configuration for the most part, it is still involved in performing config negation during UP switchover. Config negation means removing the configuration of all the other active UPs from the standby UP that is about to take over for a given active UP. So, say, in a 3:1 scenario, Active3 UP goes, down. The standby has the host-specific configurations of Active1, Active2, and Active3. Since the standby now takes over for Active3, the RCM negates the configs of Active1 and Active2 from that standby as part of the switchover.

NETCONF Notification Subscription on NSO

All notifications sent from RCM are captured by NSO. NSO filters the notifications and handles RCM related notifications.

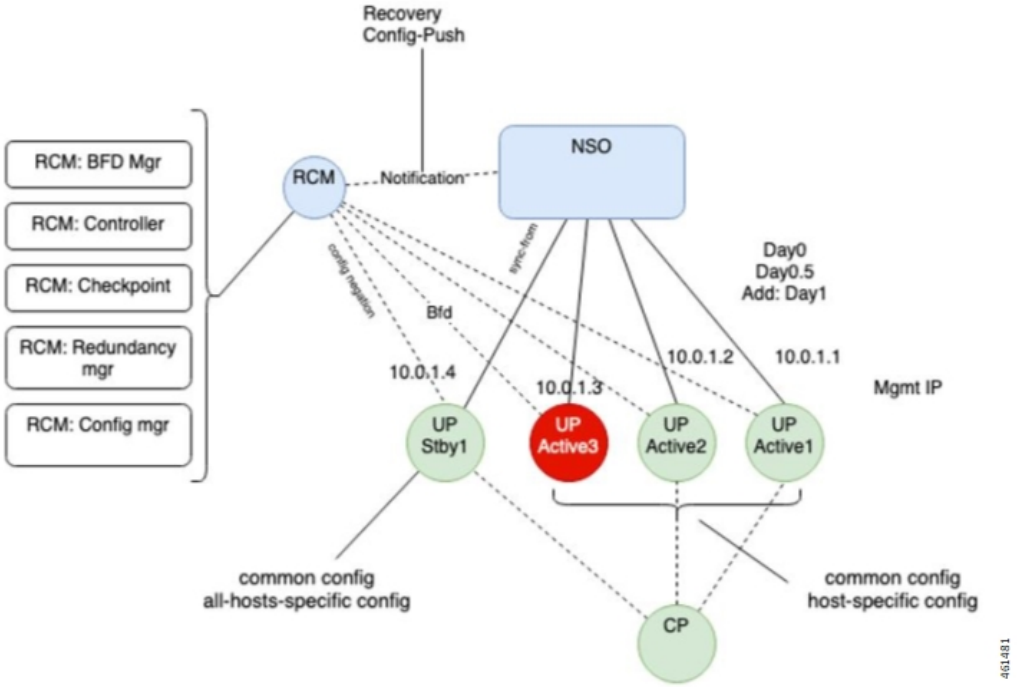
The following table explains the possible types of RCM UP notifications, and how they are handled by NSO.

	Recovery	**Config-Push**	
	UP Recovery	**UP Reboot**	**New UP**
Active UP	Update Device meta-data in NSO.	**Push host** specific config. If common config is not on device, re-push the common config file. Update device meta-data in NSO.	
Standby UP	N/A	Push **all hosts** specific config. If common config is not on device, re-push the common config file. Update device meta-data in NSO.	

Handle RCM UP Recovery Notification

In case of UP failure, RCM detects the failure via BFD Manager, and pushes the notification, which is received by NSO. RCM handles the switchover of the UP to make an elected standby UP to an active one. This configuration management process for the standby UP to switchover does not take much time because the standby UP already has all the required configuration.

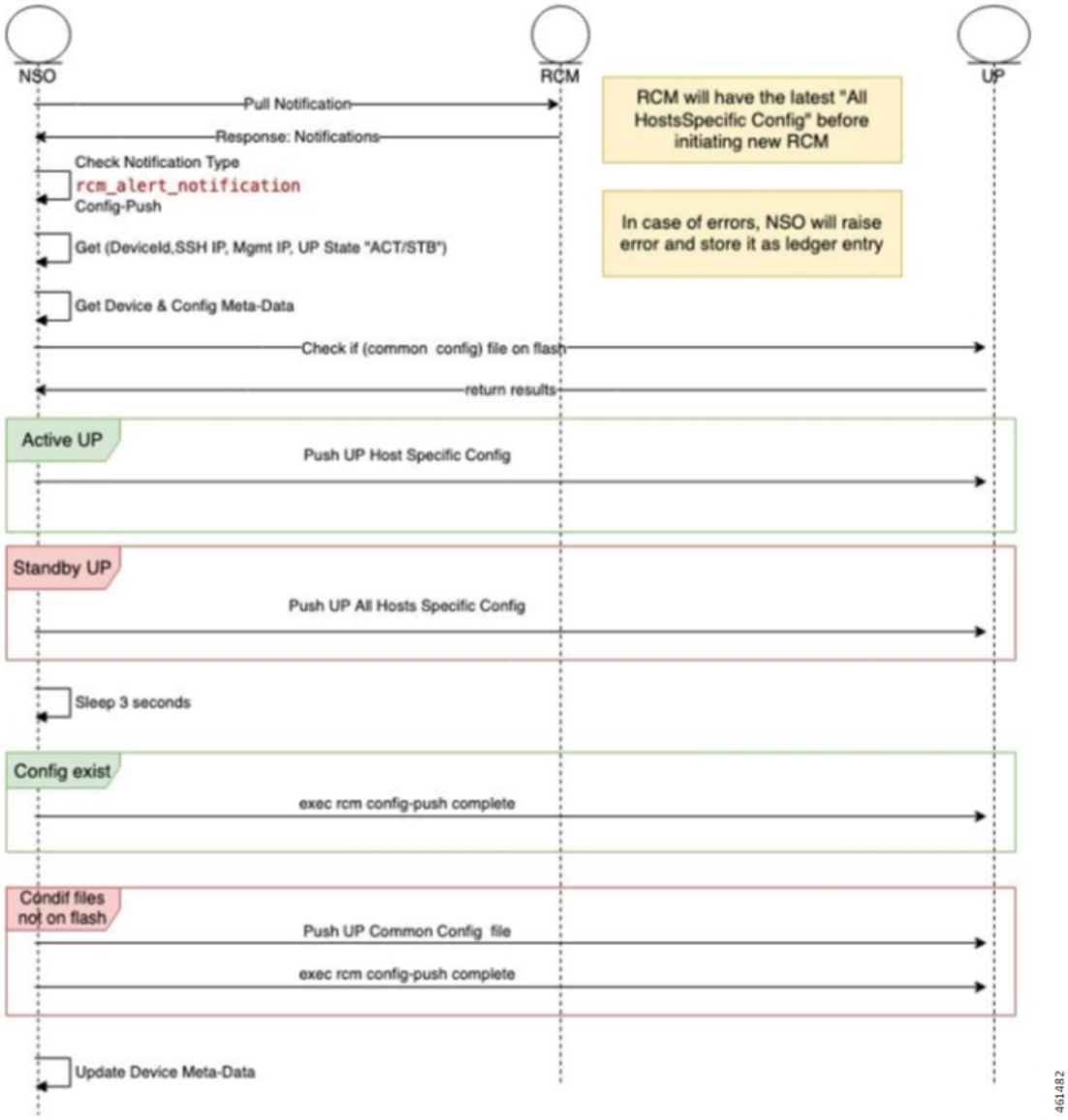
The following figure illustrates the RCM UP notification handling:



RCM UP Config-Push Notification

RCM generates config-push notification if there is a new UP coming up or existing UP is rebooting for recovery.

The following figure illustrates the RCM UP Configuration Push notification handling



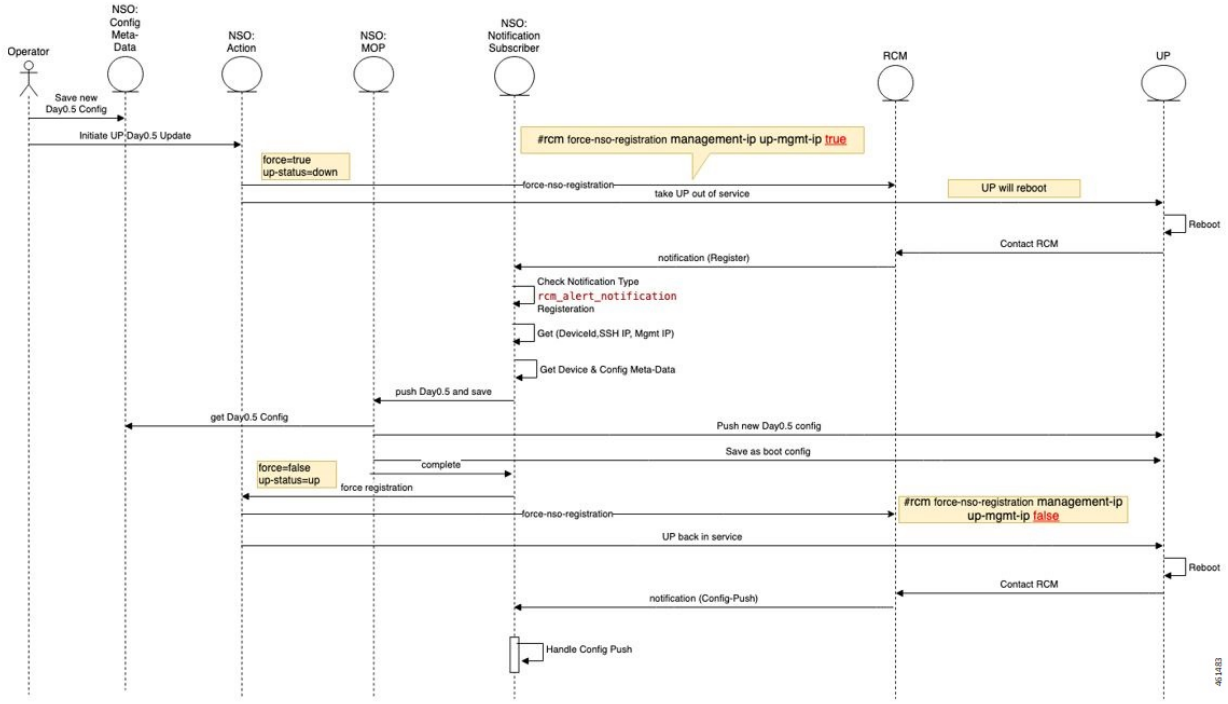
NSO performs the following steps whenever there is UP Config-Push notification:

1. Waits for alert-status “config-push”
2. Gets the device details (Device Name, SSH IP, Management IP, UP Status)
3. Gets the device meta-data from NSO
4. Checks if common file exists in UP flash.
5. If the UP State is Active, NSO pushes the UP host specific config file using the Mobility MOP.
6. If the UP State is Standby, NSO pushes the UP all hosts specific config file using the Mobility MOP.
7. Sleeps for 3 seconds.
8. If the common config file exists, NSO applies it by running live-status command on UP

9. If the common config file doesn't exist, NSO pushes the common config file to the UP using the Mobility MOP, and then applies it by running live-status command on UP.
10. Updates the device meta-data in NSO.

UP Day-0.5 Update

To change UP Day-0.5 configurations, UP must be rebooted which requires UP to be down during the change. RCM supports this use case through a command to force sending of notification whenever a specific UP gets rebooted. This notification triggers NSO to push the new Day-0.5 configurations.



1. You must update the UP Day-0.5 configuration in NSO config-metadata.
2. You must start the UP Day-0.5 change action by providing the UP device-name and management IP address.
3. NSO Action runs RCM command to force NSO registration when the UP reboots:

   `rcm force-nso-registration management-ip` *`MGMT-IP`* **`true`**
4. NSO action brings down the UP, which can be one of the following two scenarios:
   a. Standby UP: NSO action runs reload command on UP.
   b. Active UP: NSO Action runs planned switchover command on RCM by providing the UP management-ip as well as the Standby UP management-ip. The Standby UP management-ip can be retrieved from NSO Device-Metadata of the UP Group
5. After UP reboot (Active or Standby UP), RCM generates notification of type "Registration" which is received by the NSO.

6. After NSO RCM Subscriber receives the Registration notification, it starts new MOP process to push Day-0.5 configuration for the target UP.
7. NSO RCM Subscriber keeps pulling the MOP Status. After the MOP is completed, NSO runs the following command UP:

   rcm-config-push-complete
8. NSO RCM Subscriber calls the NSO Action to disable the force-notification command by running the following command:

   rcm force-nso-registration management-ip *MGMT-IP* **false**
9. NSO Action brings up the UP by running the reload command.
10. After the UP reboots, RCM generates the Config-Push notification, which is handled by NSO as usual.

Prerequisites for Configuration Push

The following are some prerequisites for config-push:

1. For UPs, the

   require rcm-configmgr

   CLI command must be preconfigured as part of Day-0 configuration. This is required for N:M, 1:1, and standalone scenarios. This enables appropriate behavior of ECS configuration.
2. For UPs, PFD push must be disabled from the CPs and in the UPs, wherever applicable. All UP configurations are pushed from the NSO directly. This is applicable for N:M, 1:1, and standalone scenarios.
3. The RCM OpsCenter Configuration mode CLIs must be configured as follows:

   k8 smf profile rcm-config-ep config-mode NSO

   k8 smf profile rcm-config-ep switchover deployment false
4. There are some default StarOS NED settings that must be overridden:
   a. Any configuration change is automatically saved to the devices boot configuration (system.cfg). This is not desirable when N:M redundancy is used as the configuration of a UP changes depends on its role. This must be disabled globally using the following CLI configuration command:

      devices global-settings ned-settings cisco-staros write-memory-setting disabled

      If using only 1:1 or standalone deployments, then this setting can be left as is.

      If using a mix of N:M and 1:1/standalone, then disable config save as above, and then fuse the "save-config-permanently" parameter in any manual configuration push for 1:1/standalone. For automated configuration pushes, the mobility function pack automatically saves the configuration whenever required.
   b. The NED treats any warnings as errors and fails the configuration push. In a lot of cases, the warnings can be ignored, and the configuration push needs to proceed. The NED can be configured to ignore select warnings using a regular expression for that warning. Here's an example.

      devices global-settings ned-settings cisco-staros behaviour config-warning-ignore .*not recommended to change the dictionary.*

Some of the other common ones are:

```
ned-settings cisco-staros behaviour config-warning-ignore .*About
to overwrite boot.*

ned-settings cisco-staros behaviour config-warning-ignore .*Standby
 card not ready.*
```

This last one is required for configuration push to SIs.

5. RCM supports the concept of an SSH IP. An SSH IP is a way to unambiguously track a given Active UP regardless of which VM is serving that function. The NSO-based solution does not use SSH IPs. However, the solution requires configuration of a dummy SSH IP. This is configured as a secondary IP address in the management interface. To avoid any errors in this configuration, the following setting is recommended as part of Day-0.5

   ```
   configure
     redundancy-configuration-module rcm rcm
       nso-ssh-ip context local interface-name local1 mask 255.255.255.224
   ```

6. Read and write operations from the NSO to the VNFs can take more or less time depending on the latency. These are tunable as shown below. Do this only if you see issues with read or write errors that are truly due to latency. Typically, default settings should suffice.

   ```
   devices global-settings read-timeout 180

   devices global-settings write-timeout 180
   ```

Limitations and Restrictions

The NSO-based Configuration Management feature has the following limitations in this release:

- Production NSO instance can run only on popular Linux flavors (for example, RedHat, Cisco Linux, Ubuntu, CentOS, and so on).
- Only Day-1 configuration is pushed for UP on RCM notifications. No other configurations are pushed.
- For pushing Day-N configuration change at a later point, you must merge that change with the Day-1 configuration files for it to be automatically pushed on an RCM notification going forward.
- If there are changes to pre-populated configuration files, they aren't pushed automatically. It's required to push them manually for all target devices. The configuration changes only for next auto-push is considered.

  For N:M UPs, the pre-populated configuration files must be preserved on the NSO (both instances if running as an HA pair) if there is at least one VNF using them.
- Only configuration commands are supported. The show CLI commands within configuration files aren't supported.
- Any configuration, to be managed from the NSO, must be understood by the corresponding NED (StarOS NED for CPs, UPs, and RCM NED for RCMs). Currently, not all StarOS configuration commands are supported—only the most used configurations in CUPS field deployments are supported. Support for any missing commands requires a newer NED.

- Support of native StarOS CLI is not 100%. While majority of the supported CLI commands are acceptable in native StarOS CLI format, there are some cases where the NSO accepts only a variant of the corresponding StarOS CLI. Such CLIs are documented in Appendix A: Incompatible StarOS Native Command Syntax, on page 524. You can use the "dry-run" functionality of the configuration MOP to detect any errors due to incompatible/unsupported CLI prior to performing a configuration push.
- A configuration push may fail if the NSO handling the request goes down during the operation. This is applicable for both manual and automated configuration push. It is also applicable for both NSO HA and standalone NSO deployments. Operator intervention may be required depending on the exact nature of the failure.
- The Day-0.5 configuration change workflow for the N:M redundancy scenario is not fully functional in this release. For this release, the workaround is to:
  1. Remove the UP from the redundancy group (making it Standby first, if it was Active)
  2. UP boots up with Day-0 and current Day-0.5 configuration
  3. Make the Day-0.5 configuration change on the UP and save it persistently as the boot configuration
  4. Add the UP back to the redundancy group. The UP registers with the RCM and the remaining configurations are pushed by NSO automatically
- In this release, the Day-N configuration push in the N:M redundancy scenario requires a prior extra step if active-charging configuration changes are pushed using the rcm-upf MOP type. It is required to manually delete the file */flash/mobility_production.cfg* on all the UPs in that redundancy group prior to invoking the MOP.
- The standard StarOS CLI NED stores certain sensitive configuration data as cleartext locally. Access to this can be restricted by using the NACM rules on the NSO. If there are additional concerns with this, contact your Cisco Account representative for a version of the NED that encrypts this sensitive data locally. Note that this encryption is specific to the NED and NSO. StarOS encrypts sensitive data on its own—the two encryptions are separate. When NSO encrypts the sensitive data locally, it decrypts it prior to transmitting it to StarOS device (it is sent over SSH, so it is encrypted in transit, but is received by StarOS CLI as cleartext).
- For the N:M redundancy scenario, RCM supports a concept of an SSH IP. The NSO-based solution does not use the SSH IP. However, for the FCS (3.0.0 and 21.25), there is a requirement to specify an SSH IP for the solution. Any address, including private, non-routable address will suffice. This address is configured as a secondary address on the management interface of the UPs. Also refer to the Prerequisites for Configuration Push, on page 521 section for a note on the SSH IP configuration requirements for the UP.

Troubleshooting

The following options are available for troubleshooting purposes:

1. Use the dashboard output for the task-id for available details. For example:

   ```
   mobility-mop:action mop-automation-status task-id 12d5fc33-2f9e-44e3-81e3-14043d4ee39d
   ```

2. In case of failures, some alarms may be raised. These can be viewed using the

   `show alarms`

CLI command.

3. Detailed logs can be viewed by examining */var/log/ncs/ncs-java-vm.log* file. However, this is oriented towards developer debugging.

Appendix A: Incompatible StarOS Native Command Syntax

This section identifies the commands that are already supported in the NED (tagged in Bold below) but not compatible with StarOS native command syntax.

Mode	Command	Comments
context xxx/ggsn-service yyy	**ip qos-dscp qci 9 af31 gtpc af41**	It accepts "ip qos-dscp qci 9 af31" and "ip qos-dscp gtpc af41" separately. When it pushes to device, it pushes as combined.
context xxx/apn yyy	apn fnetcoriolis **default max-contexts** **default cc-roaming** **default ipv6 address alloc-method** exit	It accepts "no max-contents", "no cc-roaming", and "no ipv6 addres alloc-method" and generates "default xxx" towards the device.
context xxx/crypto map yyy (ikev2-pv4)/payload zzz match ipv4	crypto map ipsec_tunnel ikev2-ipv4 keepalive interval 4 timeout 1 num-retry 4 payload mypaiload match ipv4 **default lifetime** exit	It accepts "no lifetime" and generates "default lifetime" towards the device.
active-charging service xxx/credit-control group yyy/	credit-control group cc-m2mpt quota validity-time 600 diameter reauth-blacklisted-content content-based-rar **default diameter send-ccru on-rar always** **default diameter mscc-per-ccr-update** exit	It accepts "no xxx" and generates "default xxx" towards the device.

Mode	Command	Comments
context xxx/ikev2-ikesa transform-set yyy	ikev2-ikesa transform-set transformset_li **default encryption** **default group** **default hmac** **default lifetime** **default prf** exit	It accepts "no xxx" and generates "default xxx" towards the device.
global config mode	**end** and **#exit**	rload does not accept these commands.
global config mode	snmp trap enable	Equivalent is: no snmp trap suppress
active-charging service xxx/ruledef yyy Or any command that has "!" in it	active-charging service ecs ruledef rd-webredirect-apn-sl2sfr **bearer 3gpp imsi !range imsi-pool imsi-NOREDIRECT** exit	rload does not recognize the "!" character unless it is enclosed in double-quotes or escaped with backslash.
active-charging service xxx/ruledef yyy Any URL with %NN in it	ruledef rd1 www url = http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware**%3f**id=362713555&amp exit	The %3f must be escaped with a "\"

Mode	Command	Comments
context xxx/gtpp group yyy	gtpp group sgw **no gtpp attribute node-id** **no gtpp attribute losdv** **no gtpp trigger time-limit** **no gtpp trigger tariff-time-change** **no gtpp trigger serving-node-change-limit** **no gtpp trigger inter-plmn-sgsn-change** **no gtpp trigger qos-change** **no gtpp trigger rat-change** **no gtpp trigger ms-timezone-change** **no gtpp trigger uli-change**	StarOS does not handle these commands as a default setting, whereas NED does.
context xxx/router bgp NNN	context gy router bgp 64650 neighbor 209.165.200.226 remote-as 15557 **no neighbor 209.165.200.226 capability graceful-restart**	StarOS default setting is to have graceful-restart capability. NED handles these commands the reverse way.
context xxx/ims-auth-service yyy	p-cscf table 1 row-precedence 1 ipv4-address 209.165.200.227 secondary **ipv4-address** 209.165.200.229	The highlighted keyword must be changed to ipv4-address only for the secondary IP address.
context xxx/ims-auth-service yyy	default signaling-flag	Use "no signaling-flag” instead.
context xxx/ims-auth-service yyy	default traffic-policy general-pdp-context no-matching-gates direction downlink	Use "no traffic-policy general-pdp-context no-matching-gates direction downlink" instead.
context xxx/ims-auth-service yyy	p-cscf table 1 row-precedence 1 ipv6-address 2001:860:ffff:feb6::a secondary **ipv6-address** 2001:860:ffff:feb4::9	The highlighted keyword must be changed to ipv6-address only for the secondary IP address.

Mode	Command	Comments
context xxx/hexdump-module	default file rotation volume time-stamp monitor- subscriber-file-name	Equivalent is: no file rotation volume no file time-samp no file monitor-subscriber-file-name
context xxx/hexdump-module	default file rotation volume	Equivalent is: no file rotation volume
context xxx/hexdump-module	default file time-stamp	Equivalent is: no file time-stamp
context xxx/hexdump-module	default subscriber-file-name	Equivalent is no subscriber-file-name
context xxx/hexdump-module	default hexdump transfer-mode	Equivalent is: no hexdump transfer-mode
context xxx/hexdump-module	default hexdump push-interval	Equivalent is: no hexdump push-interval
context xxx/edr-module active-charging-service	default cdr transfer-mode push via transfer-mode	Equivalent is: no cdr transfer-mode push
context xxx/session-event-module	default event transfer-mode push via transfer-mode	Equivalent is: no event transfer-mode push

Appendix B: Example Configurations for N:M Deployment with RCM

Host-specific Configuration-UP

The following are examples of host-specific configurations for two Active UPs. These are pushed to the respective UPs.

First Active UP

```
config
context EPC2
interface loop1_up1 loopback
ip address 209.165.200.225 255.255.255.224
```

```
interface loop2_up1 loopback
ip address 209.165.200.226 255.255.255.224

interface loop3_up1 loopback
ip address 209.165.200.227 255.255.255.224

interface loop4_up1 loopback
ip address 209.165.200.228 255.255.255.224

interface loop5_up1 loopback
ip address 209.165.200.229 255.255.255.224

exit
exit

context EPC2
sx-service sx_up1
instance-type userplane
bind ipv4-address 209.165.200.229
exit

exit
exit

context EPC2
gtpu-service pgw-gtpu_up1
bind ipv4-address 209.165.200.226
exit
gtpu-service saegw-sxu_up1
bind ipv4-address 209.165.200.227
exit
gtpu-service sgw-engress-gtpu_up1
bind ipv4-address 209.165.200.228
exit
gtpu-service sgw-ingress-gtpu_up1
bind ipv4-address 209.165.200.225

exit
exit
config
context EPC2
user-plane-service up_up1
associate gtpu-service pgw-gtpu_up1 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up1 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up1 sgw-egress
associate gtpu-service saegw-sxu_up1 cp-tunnel
associate sx-service sx_up1
associate fast-path service
associate control-plane-group g1
exit

exit
exit
```

Second Active UP

```
config
context EPC2
interface loop1_up2 loopback
ip address 209.165.200.230 255.255.255.224

interface loop2_up2 loopback
ip address 209.165.200.231 255.255.255.224
```

```
interface loop3_up2 loopback
ip address 209.165.200.232 255.255.255.224

interface loop4_up2 loopback
ip address 209.165.200.233 255.255.255.224

interface loop5_up2 loopback
ip address 209.165.200.234 255.255.255.224

exit
exit

context EPC2
sx-service sx_up2
instance-type userplane
bind ipv4-address 209.165.200.234
exit

exit
exit

context EPC2
gtpu-service pgw-gtpu_up2
bind ipv4-address 209.165.200.231
exit
gtpu-service saegw-sxu_up2
bind ipv4-address 209.165.200.232
exit
gtpu-service sgw-engress-gtpu_up2
bind ipv4-address 209.165.200.233
exit
gtpu-service sgw-ingress-gtpu_up2
bind ipv4-address 209.165.200.230

exit
exit

context EPC2
user-plane-service up_up2
associate gtpu-service pgw-gtpu_up2 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up2 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up2 sgw-egress
associate gtpu-service saegw-sxu_up2 cp-tunnel
associate sx-service sx_up2
associate fast-path service
associate control-plane-group g1
exit

exit
exit
```

Host-specific Configuration-RCM

The following are examples of host-specific configurations for RCM. This is pushed to the RCM.

First Active RCM

```
config
control-plane-group g1
 redundancy-group 1
  host Active1
```

```
   peer-node-id ipv4-address 209.165.200.240
  exit
 exit
exit
context EPC2
 interface-loopback loop1_up1
  redundancy-group 1
   host Active1
    ipv4-address 209.165.200.224/27
   exit
  exit
 exit
 interface-loopback loop2_up1
  redundancy-group 1
   host Active1
    ipv4-address 209.165.201.0/27
   exit
  exit
 exit
 interface-loopback loop3_up1
  redundancy-group 1
   host Active1
    ipv4-address 209.165.202.128/27
   exit
  exit
 exit
 interface-loopback loop4_up1
  redundancy-group 1
   host Active1
    ipv4-address 192.0.2.0/24
   exit
  exit
 exit
 interface-loopback loop5_up1
  redundancy-group 1
   host Active1
    ipv4-address 198.51.100.0/24
   exit
  exit
 exit
 user-plane-service up_up1
  redundancy-group 1
   host Active1
    associate control-plane-group g1
    associate fast-path service
    associate sx-service sx_up1
    associate gtpu-service pgw-gtpu_up1 pgw-ingress
    associate gtpu-service saegw-sxu_up1 cp-tunnel
    associate gtpu-service sgw-engress-gtpu_up1 sgw-egress
    associate gtpu-service sgw-ingress-gtpu_up1 sgw-ingress
   exit
  exit
 exit
 gtpu-service pgw-gtpu_up1
  redundancy-group 1
   host Active1
    bind ipv4-address 209.165.201.0
   exit
  exit
 exit
 gtpu-service saegw-sxu_up1
  redundancy-group 1
   host Active1
    bind ipv4-address 209.165.202.128
```

```
  exit
 exit
exit
gtpu-service sgw-engress-gtpu_up1
 redundancy-group 1
  host Active1
   bind ipv4-address 192.0.2.0
  exit
 exit
exit
gtpu-service sgw-ingress-gtpu_up1
 redundancy-group 1
  host Active1
   bind ipv4-address 198.51.100.123
  exit
 exit
exit
sx-service sx_up1
 redundancy-group 1
  host Active1
   bind ipv4-address 198.51.100.0
   instance-type userplane
  exit
 exit
exit
exit
```

Second Active RCM

```
config
control-plane-group g1
 redundancy-group 1
  host Active2
   peer-node-id ipv4-address 209.165.200.240
  exit
 exit
exit
context EPC2
 interface-loopback loop1_up2
  redundancy-group 1
   host Active2
    ipv4-address 209.165.200.224/27
   exit
  exit
 exit
 interface-loopback loop2_up2
  redundancy-group 1
   host Active2
    ipv4-address 209.165.201.0/27
   exit
  exit
 exit
 interface-loopback loop3_up2
  redundancy-group 1
   host Active2
    ipv4-address 209.165.202.128/27
   exit
  exit
 exit
 interface-loopback loop4_up2
  redundancy-group 1
   host Active2
    ipv4-address 192.0.2.0/24
   exit
```

```
 exit
exit
interface-loopback loop5_up2
 redundancy-group 1
  host Active2
   ipv4-address 198.51.100.0/24
  exit
 exit
exit
user-plane-service up_up2
 redundancy-group 1
  host Active2
   associate control-plane-group g1
   associate fast-path service
   associate sx-service sx_up2
   associate gtpu-service pgw-gtpu_up2 pgw-ingress
   associate gtpu-service saegw-sxu_up2 cp-tunnel
   associate gtpu-service sgw-engress-gtpu_up2 sgw-egress
   associate gtpu-service sgw-ingress-gtpu_up2 sgw-ingress
  exit
 exit
exit
gtpu-service pgw-gtpu_up2
 redundancy-group 1
  host Active2
   bind ipv4-address 209.165.201.0
  exit
 exit
exit
gtpu-service saegw-sxu_up2
 redundancy-group 1
  host Active2
   bind ipv4-address 209.165.202.128
  exit
 exit
exit
gtpu-service sgw-engress-gtpu_up2
 redundancy-group 1
  host Active2
   bind ipv4-address 192.0.2.0
  exit
 exit
exit
gtpu-service sgw-ingress-gtpu_up2
 redundancy-group 1
  host Active2
   bind ipv4-address 198.51.100.123
  exit
 exit
exit
sx-service sx_up2
 redundancy-group 1
  host Active2
   bind ipv4-address 198.51.100.0
   instance-type userplane
  exit
 exit
exit
exit
```

Common Configuration

The following is an example of a common configuration. This is pushed to all Active UPs and all Standby UPs.

```
config
 active-charging service ACS
 idle-timeout udp 60
 statistics-collection ruledef all
 host-pool IPv6_VoLTE_Phone_Host_7
 ip 209.165.200.224/27
 ip 64:ff9b::d3f6:6b00/120
 ip 2001:e60:6000::/46
 ip 2001:e60:6004::/46
 ip range 209.165.200.225 to 209.165.200.234
 ip range 64:ff9b::e00:4f12 to 64:ff9b::e00:4f14
 ip range 64:ff9b::3d6e:ff52 to 64:ff9b::3d6e:ff59
 ip range 64:ff9b::d3f6:682c to 64:ff9b::d3f6:683e
 exit
 port-map M_learning_Port
 port range 1 to 9500
 port range 10001 to 30000
 exit
 port-map OTM_Advertisement_port
 port 90
 port 9090
 exit
 ruledef ICMP
 ip protocol = 1
 exit
 ruledef ICMPv6
 ip protocol = icmpv6
 exit
 ruledef IPv6_VoLTE_Phone_1
 udp either-port range port-map M_learning_Port
 ip server-ip-address range host-pool IPv6_VoLTE_Phone_Host_7
 exit
 ruledef RD-allTraffic
 ip any-match = TRUE
 exit
 ruledef RD_Charge
 ip server-ip-address = 209.165.201.0/27
 exit
 ruledef catchall
 ip any-match = TRUE
 exit
 ruledef googles
 icmpv6 any-match = TRUE
 exit
 ruledef qci1
 tcp any-match = TRUE
 exit
 ruledef route-ims-ipv6-nexthop
 ip uplink = TRUE
 ip version = ipv6
 exit
 ruledef optIn
 ip any-match = TRUE
 exit
 group-of-ruledefs GoR_FOTA
 add-ruledef priority 1 ruledef FOTA_SAMSUNG
 add-ruledef priority 2 ruledef FOTA_LG
 add-ruledef priority 3 ruledef FOTA_LG_2
```

```
 add-ruledef priority 5 ruledef FOTA_PANTECH_2
 add-ruledef priority 8 ruledef IOS_OTA_Update
 add-ruledef priority 9 ruledef GOTA_google
 add-ruledef priority 10 ruledef FOTA_Hybrid_Egg
 add-ruledef priority 11 ruledef FOTA_SAMSUNG_2
 add-ruledef priority 12 ruledef FOTA_SAMSUNG_3
 add-ruledef priority 13 ruledef FOTA_SAMSUNG_4
 add-ruledef priority 15 ruledef FOTA_SAMSUNG_5
 add-ruledef priority 16 ruledef FOTA_LG_4
 add-ruledef priority 17 ruledef FOTA_LG_5
 add-ruledef priority 18 ruledef FOTA_HUAWEI_Egg
 add-ruledef priority 20 ruledef KTF_DMS_FOTA
 add-ruledef priority 21 ruledef FOTA_Nlabs
 add-ruledef priority 22 ruledef FOTA_LTE_Beam
 add-ruledef priority 23 ruledef FOTA_S_Mobile
 add-ruledef priority 24 ruledef FOTA_Giga_Genie
 add-ruledef priority 100 ruledef SAMSUNG_SKT_issue
 add-ruledef priority 104 ruledef new_FOTA_Pantech
 add-ruledef priority 106 ruledef new_FOTA_KTtech
 add-ruledef priority 107 ruledef new_IOS_OTA_Log
 add-ruledef priority 114 ruledef new_FOTA_LG_3
 add-ruledef priority 200 ruledef IoT_FOTA_mexus
 add-ruledef priority 201 ruledef IoT_FOTA_acnt
 add-ruledef priority 202 ruledef IoT_FOTA_amtel
 exit
 packet-filter qci1
 ip protocol = 1
 ip remote-port = 1001
 priority 1
 exit
 packet-filter subscriber-pools
 exit
 charging-action CA-nothing
 content-id 5
 exit
 charging-action CA_Chargeable_2
 content-id 1
 billing-action egcdr
 exit
charging-action CA_Charge
exit
 charging-action DSI
 billing-action egcdr
 flow action discard
 tft packet-filter permit_all
 exit
 charging-action ca11
 service-identifier 22
 billing-action egcdr
 cca charging credit
 flow action discard
 flow limit-for-bandwidth id 4
 exit
 charging-action catchall
 content-id 10
 billing-action egcdr
 cca charging credit rating-group 10 preemptively-request
 exit
 charging-action qci1
 billing-action egcdr
 cca charging credit rating-group 1 preemptively-request
 qos-class-identifier 1
 tft packet-filter qci1
 exit
```

```
 bandwidth-policy bw-policy
 flow limit-for-bandwidth id 2 group-id 2
 flow limit-for-bandwidth id 4 group-id 4
 flow limit-for-bandwidth id 10 group-id 12
 flow limit-for-bandwidth id 562 group-id 562
 group-id 2 direction downlink peak-data-rate 225280 peak-burst-size 2253 violate-action
discard
 group-id 4 direction uplink peak-data-rate 450560 peak-burst-size 4506 violate-action
discard
 group-id 10 direction uplink peak-data-rate 1153434 peak-burst-size 11534 violate-action
discard
 group-id 11 direction uplink peak-data-rate 10000 peak-burst-size 10000 violate-action
discard
 exit
 rulebase 5G-DF
 tcp packets-out-of-order timeout 30000
 no retransmissions-counted
 edr sn-charge-volume count-dropped-units
 bandwidth default-policy bw-policy
 exit
 rulebase RB-allTraffic
 action priority 100 ruledef RD-allTraffic charging-action CA_Charge
 egcdr threshold interval 3600
 egcdr threshold volume total 4000000
 exit
 rulebase RB_Charge
 action priority 10 ruledef RD_Charge charging-action CA_Charge
 exit
 rulebase cisco
 billing-records egcdr
 action priority 12 ruledef catchall charging-action catchall monitoring-key 123
 egcdr threshold interval 120
 egcdr threshold volume total 1000000
 exit
 rulebase cisco_dynamic
 action priority 11 dynamic-only ruledef qci1 charging-action qci1
 action priority 10000 ruledef catchall charging-action catchall
 egcdr threshold interval 120
 egcdr threshold volume total 100000
 exit
 rulebase P2P
 transactional-rule-matching
 dynamic-rule order first-if-tied
 tethering-detection application ip-ttl value 62
 flow end-condition timeout normal-end-signaling session-end charging-edr flow-edr
 billing-records egcdr
 edr transaction-complete http charging-edr http-edr
 flow control-handshaking charge-to-application all-packets
 egcdr threshold interval 3600
 egcdr threshold volume total 4000000000
 no cca quota retry-time
 cca diameter requested-service-unit sub-avp volume cc-total-octets 5000
 p2p dynamic-flow-detection
 no tft-notify-ue-def-bearer
 exit
 rulebase default
 exit
 rulebase wap_adult
      transactional-rule-matching
      tcp mss 1320 limit-if-present
      flow end-condition handoff timeout normal-end-signaling session-end charging-edr
flow-edr
      billing-records egcdr radius
      action priority 28 ruledef catchall charging-action CA_Chargeable_2
```

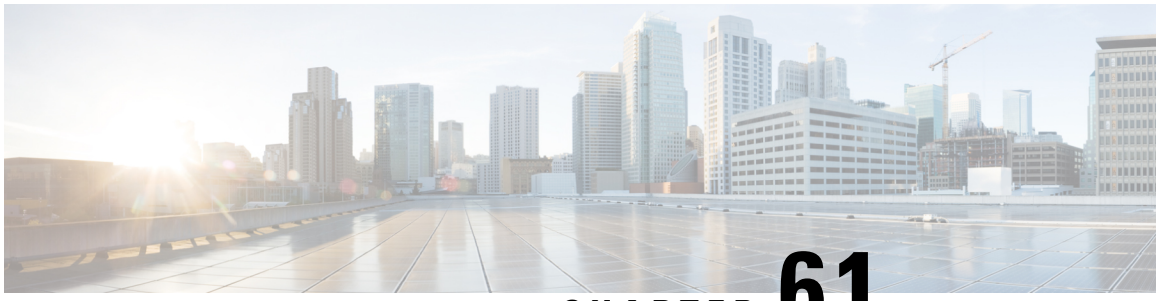
```
      action priority 29 ruledef catchall charging-action CA_Chargeable_2
      edr transaction-complete http charging-edr http-edr
     flow control-handshaking charge-to-application  mid-session-packets tear-down-packets

      egcdr threshold volume total 3000000
#exit
 service-scheme ss1
 exit
 credit-control group DCCA_grp1
 diameter origin endpoint Gy
 diameter peer-select peer minid-Gy
 pending-traffic-treatment noquota buffer
 pending-traffic-treatment quota-exhausted buffer
 pending-traffic-treatment validity-expired pass
 exit
 credit-control group default
 pending-traffic-treatment noquota pass
 pending-traffic-treatment quota-exhausted buffer
 exit
 policy-control charging-rule-base-name active-charging-rulebase
 policy-control burst-size auto-readjust duration 3
 exit
 context ecs
 apn cisco.com
 ip context-name ecs
 exit
 apn starent.com
 ip context-name ecs
 exit
end
```

Standby Configuration (Active1 + Active2)

```
config
context EPC2
interface loop1_up1 loopback
ip address 198.51.100.123 255.255.255.224

interface loop2_up1 loopback
ip address 209.165.201.0 255.255.255.224

interface loop3_up1 loopback
ip address 209.165.202.128 255.255.255.224

interface loop4_up1 loopback
ip address 192.0.2.0 255.255.255.224

interface loop5_up1 loopback
ip address 198.51.100.0 255.255.255.224

exit
exit

context EPC2
sx-service sx_up1
instance-type userplane
bind ipv4-address 198.51.100.0
exit

exit
exit

context EPC2
```

```
gtpu-service pgw-gtpu_up1
bind ipv4-address 209.165.201.0
exit
gtpu-service saegw-sxu_up1
bind ipv4-address 209.165.202.128
exit
gtpu-service sgw-engress-gtpu_up1
bind ipv4-address 192.0.2.0
exit
gtpu-service sgw-ingress-gtpu_up1
bind ipv4-address 198.51.100.0

exit
exit

context EPC2
user-plane-service up_up1
associate gtpu-service pgw-gtpu_up1 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up1 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up1 sgw-egress
associate gtpu-service saegw-sxu_up1 cp-tunnel
associate sx-service sx_up1
associate fast-path service
associate control-plane-group g1
exit

exit
exit

config
context EPC2
interface loop1_up2 loopback
ip address 209.165.200.230 255.255.255.224

interface loop2_up2 loopback
ip address 209.165.200.231 255.255.255.224

interface loop3_up2 loopback
ip address 209.165.200.232 255.255.255.224

interface loop4_up2 loopback
ip address 209.165.200.233 255.255.255.224

interface loop5_up2 loopback
ip address 209.165.200.234 255.255.255.224

exit
exit

context EPC2
sx-service sx_up2
instance-type userplane
bind ipv4-address 209.165.200.234
exit

exit
exit

context EPC2
gtpu-service pgw-gtpu_up2
bind ipv4-address 209.165.200.231
exit
gtpu-service saegw-sxu_up2
bind ipv4-address 209.165.200.232
```

```
exit
gtpu-service sgw-engress-gtpu_up2
bind ipv4-address 209.165.200.233
exit
gtpu-service sgw-ingress-gtpu_up2
bind ipv4-address 209.165.200.230

exit
exit

context EPC2
user-plane-service up_up2
associate gtpu-service pgw-gtpu_up2 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up2 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up2 sgw-egress
associate gtpu-service saegw-sxu_up2 cp-tunnel
associate sx-service sx_up2
associate fast-path service
associate control-plane-group g1
exit

exit
exit
```



CHAPTER 61

NSO Orchestration for 4G CUPS

- Revision History, on page 539
- Feature Description, on page 539
- Use Cases, on page 539
- How it Works, on page 540
- Installing NSO Packages, on page 547
- VNF Orchestration/Deployment and Automatic Configuration Management, on page 548
- Appendix A: YANG definition of VNF, on page 569

Revision History

Revision Details	Release
First introduced.	21.25

Feature Description

The Cisco Network Service Orchestrator (NSO) based VNF orchestration enables you to manage the lifecycle of newly created Virtual Network Function (VNF) devices such as CP, UP, and RCM.

The Cisco NSO Orchestration for 4G CUPS solution provides the following functions:

- Instantiation via NSO CLI, Web-Interface, or NSO RESTCONF API
- Onboarding of VNF devices such as CP, UP, and RCM upon successful instantiation
- Pushing of Day-0.5, and Day-1 CUPS configuration after successful instantiation
- Decommission of the VNF devices

Use Cases

The NSO orchestration solution caters to the following use cases:

1. Instantiation of new CP, UP, and RCM

Instantiating new 4G-based VNFs (CPs, UPs, or RCMs) for CUPS. CP can be a Virtualized Packet Core-Single Instance (VPC-SI) or Virtualized Packet Core-Distributed Instance (VPC-DI), but UP can only be a VPC-SI.

Users are notified if there are any failures.

2. Termination of CP, UP, and RCM

   Terminating 4G-based VNFs (CPs, UPs, or RCMs) for CUPS.

   Users are notified if there are any failures.

3. Updating current status on the VNF dashboard

   Providing the current status on the dashboard of VNFs.

4. Configuration of logical group for CP, UPs, and RCMs

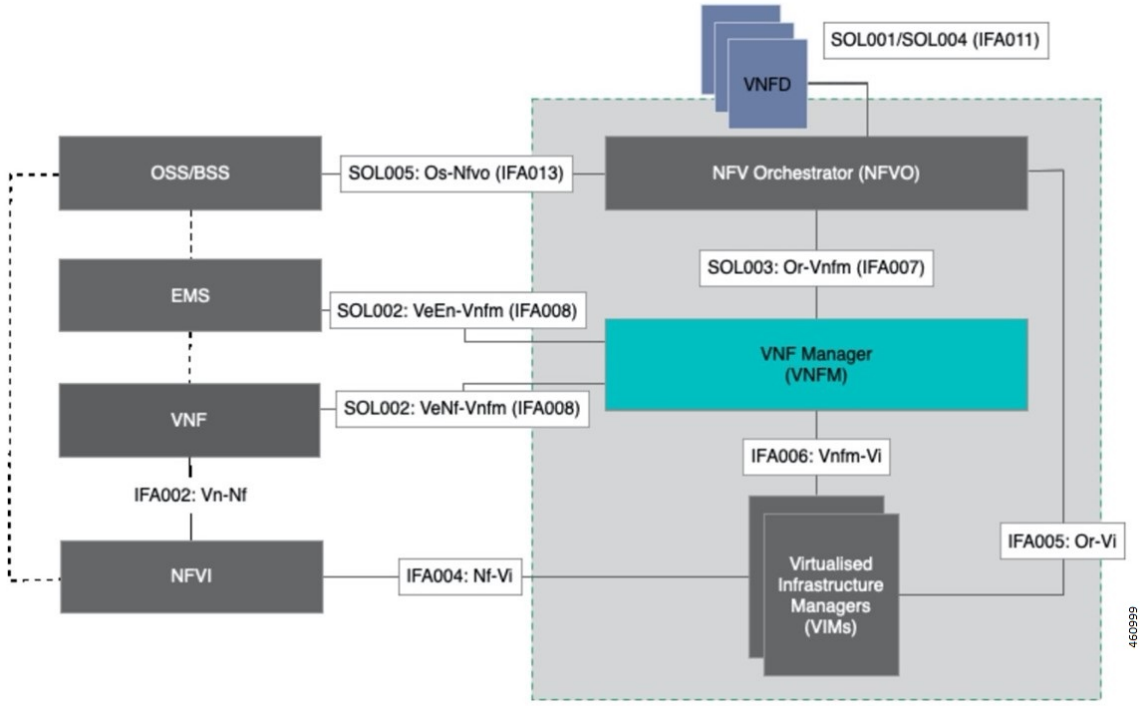
   Configuring device group in NSO to group the CP, UPs, and RCMs, and adding the corresponding VNFs to the device group.

How it Works

Architecture

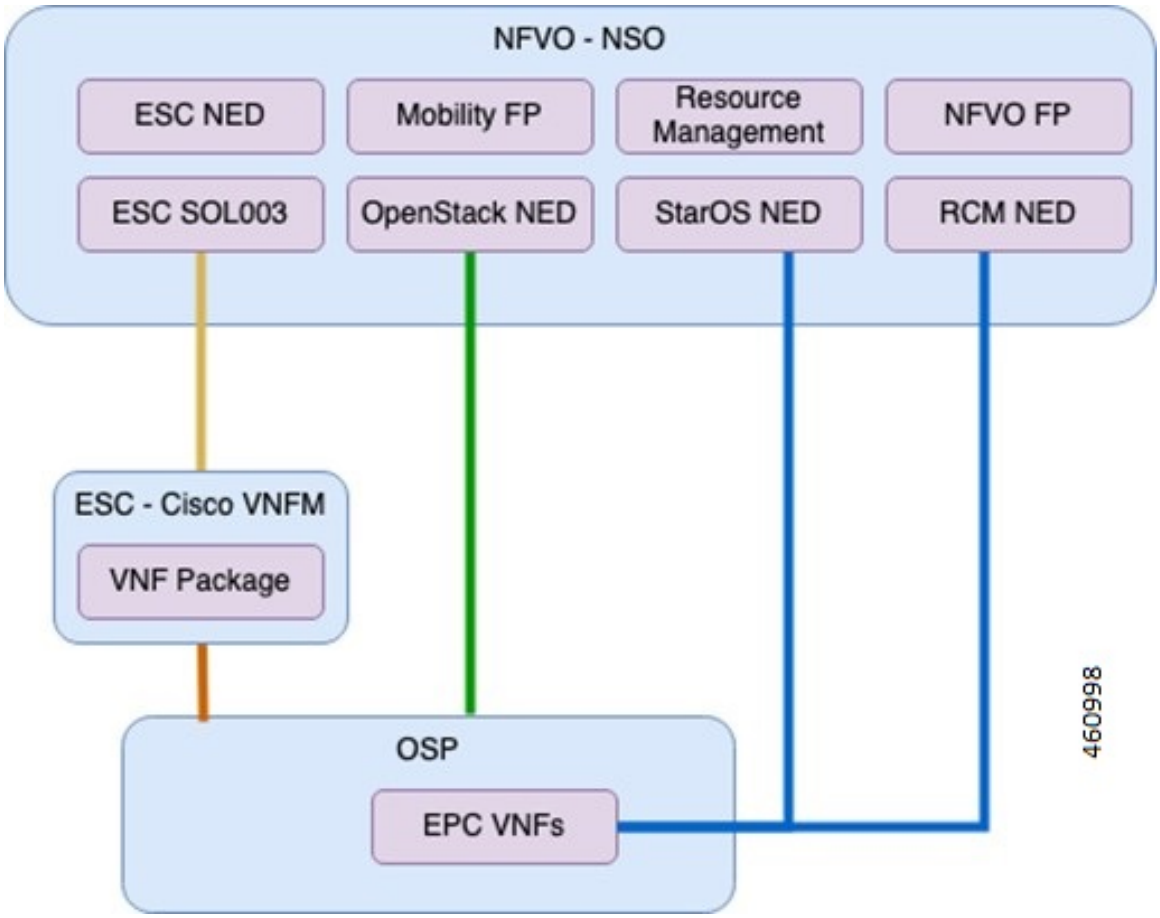
The Cisco NSO orchestration engine software modules handle the Network Functions Virtualization Orchestrator (NFVO) functions. The NFV solution follows the ETSI NFV Management and Orchestration (MANO) model, as shown in the following figure.

Figure 32: NFV Solution Architecture



The following diagram illustrates, at a high-level, the components and frameworks involved in the solution.

Figure 33: NFV Solution Components



Components

The following are some of the important components of NSO:

- **Cisco NFVO Functional Pack**:

  Cisco NFVO Functional Pack contains the YANG models according to the MANO specification (SOL006).

  Cisco NFVO Functional Pack contains models for **cisco-etsi-nfvo**, which implements the instantiation logic of MANO descriptors on VNF Managers (VNFM) and OpenStack. Virtual Network Function (VNF) and Network Service (NS) are the main services in this package. Northbound users interact with these services to start VNFs or network services.

  It also includes models for **cisco-etsi-nfvo-ro**, which contains the Resource Orchestration (RO) functionality. Resource Orchestration manages the allocation of physical resources in the Virtualized Infrastructure Managers (VIMs). These physical resources are used by a VNF or an NS.

- **StarOS NED for NSO**:

  StarOS-based Network Element Driver (NED) interfaces with the Cisco 4G CUPS VNFs for configuration push.

- **RCM NED for NSO**:

RCM-based NETCONF NED is used to establish communication between NSO and RCM devices.

- **Cisco ESC SOL003 NED**:

  This NED is used for ETSI SOL3 compliant devices. Elastic Services Controller (ESC) is also added as SOL3 compliant device to NSO.

- **NFV Apps Mobility Package:**

  This is a custom package that provides VNF life-cycle management, and VNF dashboard update.

Minimum Platform and Software Requirements

The following are the minimum platform and software requirements to support NSO Orchestration:

- Supported VIM: OpenStack
- Supported VNFM: Cisco ESC
- Supported Orchestrator: NSO
- Network Elements:
  - RCM
  - VPC-SI (UP/CP)
  - VPC-DI (CP)

Table 36: Software Versions

Software	Minimum version
Redhat OpenStack	13 (Queens) **Note** VMWare or OSP 16 isn't supported or validated.
Cisco ESC	5.5.0
Cisco NSO	5.5.2
OpenStack NED	4.2.23
ESC NED	5.3.0.94
StarOS NSO NED	5.38.1
Cisco NFVO FP	4.6.0
Mobility FP	3.0.0
NSO Resource Management	3.5.2
Cisco NSO HCC	5.0.0

This feature supports the following ETSI MANO specifications:

Table 37: ETSI MANO Specifications

Specification	Supported Version	Description
SOL001	v2.5.1	Defines the format and structure for the VNF Descriptor
SOL003	v2.4.1	Defines all interactions over the Or-Vnfm reference point

Network and Hardware Requirements

Network Requirements:

The following table demonstrates the NSO and ESC network requirements:

Table 38: NSO and ESC Network Requirements

Application	Management IP	Orchestration	Connection between HA Pair
NSO (2 VMs + VIP)	3	3	L2 connection of 100 Mbps with latency less than 30 ms
ESC (2 VMs + VIP)	3	3	L2 connection of 100 Mbps with latency less than 30 ms

Hardware Requirements

The following table demonstrates the specifications for NSO and ESC Virtual Machine to support maximum of 250 VNFs.

Table 39: NSO and ESC VM Specifications

Application	Number of VMs	VM CPU Cores	VM RAM	VM Storage	VM Connectivity
NSO	2	8	16 GB RAM baseline + 10 MB RAM for every StarOS device to be supported	100 GB disk (preferably SSD)	One 10 Gbps network link
ESC	2	4	16 GB	100 GB	

Licensing

The NSO Orchestration for 4G CUPS is a licensed Cisco feature. Contact your Cisco Account representative for detailed information on specific licensing requirements.

Call Flows

This section describes the key call flows for the 4G CUPS orchestration functionality.

VNF Onboarding

This section describes the VNF Onboarding flow.

Figure 34: VNF Onboarding

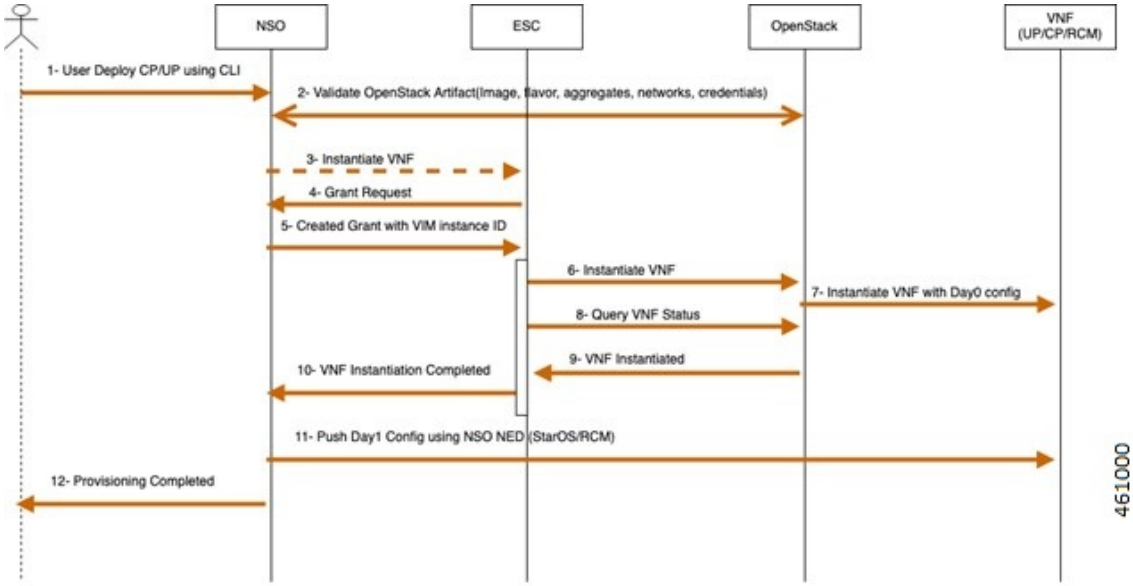


Table 40: Call Flow Description

Step	Description
1	Network operator uses NSO CLI to instantiate a VNF (CP, UP, or RCM). This includes the VIM ID to host the VNF, and the ESC.
2	NSO validates the data provided by the user via OpenStack.
3	NSO sends a SOL.003 request to instantiate the VNF on the ESC.
4	ESC sends a Grant Request to the NSO.
5	NSO sends a resource grant message to the ESC with VIM InstanceId.
6	ESC uses OpenStack API to instantiate the VNF.
7	OpenStack brings up the VNF.
8	ESC queries the OpenStack about the VNF status.
9	OpenStack replies with VNF-Up message.
10	ESC notifies the NSO about VNF instantiation.
11	NSO pushes Day-1 configuration onto the VNF.
12	NSO notifies the Operator that the VNF provisioning is complete.

P2P Module Installation

The mobility function pack supports installing a P2P module as part of VNF deployment. The P2P module is installed after the device is onboarded. The P2P module file must be uploaded to the NSO prior to the VNF deployment. The configurable parameters indicate whether P2P installation is required, and the file location.

VNF Termination

This section describes the VNF Termination flow.

Figure 35: VNF Termination Flow

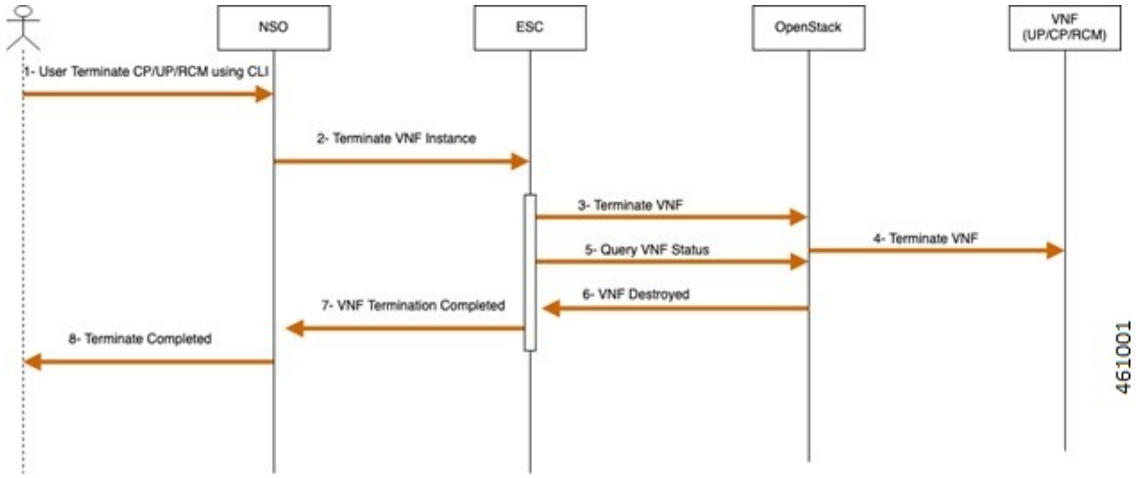


Table 41: Call Flow Description

Steps	Description
1	Operator uses NSO CLI to terminate a VNF (CP, UP, or RCM). This includes the VNF ID, VIM ID to host the VNF, and the ESC.
2	NSO sends a SOL.003 request to terminate the VNF on the ESC.
3	ESC uses OpenStack API to terminate the VNF.
4	OpenStack terminates the VNF.
5	ESC queries the OpenStack about the VNF status.
6	OpenStack replies with a VNF Destroyed message.
7	ESC notifies the NSO about VNF termination.
8	NSO notifies the Operator that the VNF termination is complete.

Recovery

Auto-healing isn’t supported currently.

To recover from fault state to previous state, perform the following steps:

- Cancel or terminate the VNF instantiation. The system returns to its original state
- Cancel or recreate VNF termination process. The system returns to its original state

Limitation

The NSO Orchestration for 4G CUPS feature has the following limitation in this release:

- Production NSO instance can run only on popular Linux flavors (for example, RedHat, Cisco Linux, Ubuntu, CentOS, and so on).
- A VNF deployment may fail if the NSO/ESC instance handling the deployment goes down. This is applicable for both ESC/NSO HA as well as standalone ESC/NSO deployments. Operator intervention is required depending on the exact nature of the failure. In case of deployment followed by automated configuration push, it's possible that the deployment succeeds but the subsequent configuration push fails depending on the timing of the NSO failure.

Installing NSO Packages

The NSO Orchestration solution uses a collection of NEDs and other NSO packages. The following is a detailed list of various packages and their roles. For installation instructions of these packages, see the "Packages" chapter in the *NSO Administration Guide* for the appropriate NSO version.

1. **NSO NED packages:**

   Most of the NSO NED packages are published for downloading independently. Contact your Cisco representative for the details on how to download.

   cisco-rcm-1.0.tar.gz - RCM NETCONF-based NED for RCM device communication from NSO

   ncs-5.5.2-cisco-staros-5.38.tar.gz - CLI-based NED for StarOS device (SI or DI) communication from NSO

   ncs-5.5.1-etsi-sol003-1.13.14.tar.gz - ETSI SOL003 based NED for ESC communication from NSO

   ncs-5.5.1-openstack-cos-4.2.23.tar.gz - Openstack NED for Openstack communication from NSO

   ncs-5.5.2.3-cisco-etsi-nfvo-4.6.0.tar.gz - NETCONF-based NED for ESC communication from NSO

   ncs-5.5.2.3-esc-5.5.0.57.tar.gz - ETSI SOL-based NED for ESC communication from NSO

2. **NSO Custom packages:**

   These are custom-built packages for Mobility VNF orchestration. NSO custom packages are bundled in the mobility function pack tar archive.

   mobility-common.tar.gz - Common package for config and device metadata

   nfv-common.tar.gz - Common packages for VNF orchestration-related common utilities

   nfv-device-onboarding.tar.gz - Package to support NSO device onboarding

   nfv-vim.tar.gz - Package for Openstack related precheck functionality

   nfv-vnf-lcm.tar.gz - Package for VNF Instantiation and termination logic

mop-common.tar.gz - Common packages for config MOP-related common utilities

mobility-mop.tar.gz - Package for Mobility MOP Config Push

3. **VNF Packages required for Orchestration (SOL003/SOL004):**

   These are VNF packages that are used for onboarding a specific VNF. These packages are provided only as guidelines. Mostly, a given package is customized to suit the deployment environment.

   VPC-SI-2P-IMAGE-BOOT - Reference SOL003/SOL004 CSAR package for SI instantiation

   RCM-IMAGE-BOOT - Reference SOL003/SOL004 CSAR package for RCM instantiation

   VPC-DI-2P-1DI-ENCRYPTVOLBOOT - Reference SOL003/SOL004 CSAR package for VPC DI instantiation with two CF and four SF. SF has two service networks.

   VPC-DI-2P-1DI-ENCRYPTVOLBOOT-LTD - Reference SOL003/SOL004 CSAR package for VPC DI instantiation with two CF and two SF. SF has two service networks.

   VPC-DI-2P-1DI-ENCRYPTVOLBOOT-LTD-1S-NETWORK - Reference SOL003/SOL004 CSAR package for VPC DI instantiation with two CF and two SF. SF has only one service network.

   create-zip.sh - Shell script to rebuild the SOL003 package, if there are any changes to SOL001 definitions or Day-0 scripts.

VNF Orchestration/Deployment and Automatic Configuration Management

This solution includes the following tasks:

- Pre-population of config metadata for VNF orchestration.
- Orchestration/Deployment of VNFs (CP, UP, or RCM)
- Automatic device onboarding post VNF deployment
- Post-deployment automatic configuration push

Pre-population of Config Metadata for VNF Orchestration

Pre-population of Config Metadata is important to achieve any post-deployment configuration push from NSO in an automated mode. If there are no prepopulated data for this device, NSO instantiates the VNF and on-boards as a device in NSO.

Prepopulating of config metadata has the following structure, and population of this data is based on the network scheme and data set:

```
container
metadata-store {
         list config-metadata {
             key device-name;
             leaf device-name {
                 tailf:info "onboarding device name";
                 type string;
             }
             leaf redundancy_scheme {
```

```
                tailf:info "cluster-topology 1:1, N:M and N+2";
                type string;
            }
            leaf device-type {
                tailf:info "Onboarding device type vpc or rcm";
                type string;
            }
            list attributes {
                key attribute-name;
                leaf attribute-name {
                    tailf:info "Attribute Name";
                    type string;
                }
                leaf attribute-value {
                    tailf:info "Attribute Value";
                    type string;
                }
            }
            list configuration-type {
                key config-type;
                tailf:info "Configuration type Day0.5, Day1 or DayN";
                leaf config-type {
                    type string;
                }
                list files {
                    key file-name;
                    tailf:info "file name";
                    leaf file-name {
                        type string;
                    }
                    leaf config-scheme {
                        type string;
                    }
                    // CP device info
                    list additional-files {
                        key device;
                        //cp device
                        leaf device {
                            tailf:info "device name";
                            type string;
                        }
                        list additional-file {
                            key additional-file-name;
                            leaf additional-file-name {
                                tailf:info "file name";
                                type string;
                            }
                        }
                    }
                }
            }
        }
    }
```

The following table provides a description of the parameters:

Parameter	Description
device-name	Name of the NSO device corresponding to the VNF. Same as VNF name.
redundancy_scheme	Type of redundancy scheme. N + 2 is standalone (no redundancy)
device-type	vpc (for SI and DI) or rcm (for RCM)

Parameter	Description
configuration-type	Day-0.5 is a special configuration for N:M redundancy. This configuration enables the UP to contact the RCM. This configuration is expected to be saved persistently.
	Day-1 is the bulk of the configuration
	Day-N generally changes to a working configuration. Does not apply to NSO Orchestration flows.
file-name	Primary configuration file(s) to be pushed
config-scheme	This parameter can have one of the following values: **Common**: Configuration is pushed to all UPs regardless of role (Active or Standby). **host-specific**: This scheme is similar to “Common”, as the configuration is pushed to all UPs (Active or Standby). However, the configuration is pushed only after “common” configuration. This enables you to provide any configuration that is dependent on the “common” configuration. For example, the control-plane group configuration. **allHostSpecific**: Contains the union of host-specific configurations for all active UPs. The configuration is pushed to all the standby UPs for N:M. **"Active1", "Active2",... "ActiveN”**: Host-specific configuration for the respective active UP. It is pushed only to the specific UP. **"Active1-rcm", "Active2-rcm", .. "ActiveN-rcm"**: This configuration is in RCM format, and is pushed to the RCMs. RCM needs this scheme to perform configuration negation when a standby takes over for a specific active device.
additional-files	This parameter pushes the related configuration to other devices (for example, pushing configuration to CP when onboarding UP). This is not yet supported.
attribute-name	This parameter identifies any attribute (variables) in the config files for dynamic substitution. Formatted as $attribute_name
attribute-value	Value for the attribute

The following is an example of NSO action to populate or modify the config meta-data:

```
container
  config-metadata {
         // config true;
         tailf:action config-metadata-request {
             tailf:info "Invoke upgrade action on the selected devices";
             tailf:actionpoint config-metadata-request;
             input {
                 list config-metadata {
                     key device-name;
                     leaf device-name {
                         tailf:info "onboarding device name";
                         type string;
                     }
                     leaf device-type {
                         tailf:info "Onboarding device type vpc or rcm";
                         type enumeration {
                             enum vpc;
                             enum rcm;
                         }
                     }
                     leaf redundancy_scheme {
```

```
    tailf:info "cluster-topology 1:1, N:M and N+2";
    type enumeration {
        enum 1:1;
        enum N:M;
        enum RCUPS;
    }
}


list configuration-type {
    key config-type;
    tailf:info "Configuration type Day0.5, Day1 or DayN";
    leaf config-type {
        type enumeration {
            enum Day0.5;
            enum Day1;
            enum DayN;
        }
    }
    list files {
        key file-name;
        tailf:info "file name";
        leaf file-name {
            type string;
        }
        leaf config-scheme {
            type enumeration {
                enum common;
                enum host-specific;
                enum host-specific-common;
            }
        }
        // CP device info
        list additional-files {
            key device;
            //cp device
            leaf device {
                tailf:info "device name";
                type string;
            }
            list additional-file {
                key additional-file-name;
                leaf additional-file-name {
                    tailf:info "file name";
                    type string;
                }
            }


            }
        }
    }
}
list attributes {
    key attribute-name;
    leaf attribute-name {
        tailf:info "Attribute Name";
        type string;
    }
    leaf attribute-value {
        tailf:info "Attribute Value";
        type string;
    }
}
```

```
                }
                leaf delete-config-data {
                    type boolean;
                    default false;
                }
            }
            output {
                leaf status {
                    type string;
                }
                leaf message {
                 type string;
                }
            }
        }
    }
```

You can call this action from RESTCONF, as shown in the following example:

URI:
http://<NSO-IP>:<NSO-REST-PORT>/restconf/data/mobility-common:config-metadata/config-metadata-request

Method: POST

Content-Type: application/yang-data+json

Payload:

```
{
        "config-metadata": {
            "device-name": "test2",
     "schema" : "1:1",
            "attributes":{
                "attribute-name":"test",
                 "attribute-value": "gh"
                  },
            "configuration-type":{
                "config-type": "Day0.5",
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/test.txt"
                    },
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/day0.5.txt"
                    }

            }
        }
}
```

Result:

```
{
    "mobility-common:output": {
        "status": "Success
/home/ubuntu/tmo_action/test.txt ==> syntax error: unknown command,Error: on line 3: kkkl,
/home/ubuntu/tmo_action/day0.5.txt ==> Success"
    }
}
```

You can call this action using NCS CLI, as shown in the following example:

```
ubuntu@ncs> request config-metadata config-metadata-request config-metadata { device-name
staros-1 attributes { attribute-name hostname attribute-value TEST } configuration-type {
config-type Day0.5 files { file-name /home/ubuntu/tmo_action/test.txt } files { file-name
/home/ubuntu/tmo_action/day0.5.txt } } schema 1:1  }
status Success
```

```
/home/ubuntu/tmo_action/test.txt ==> syntax error: unknown command,Error: on line 3: kkkl,
/home/ubuntu/tmo_action/day0.5.txt ==> Success
[ok][2021-07-12 08:05:01]
```

NOTES:

- Config-metadata-request action has internal config validator. Config validator allows detection of syntax or certain semantic errors (for example, out of range values) in advance before pushing the configuration. Config validation requires at least a device which is onboarded in NSO (Either real-one or NetSim).

  The configurable parameter is as follows:

  ```
  container
  configurable-parameters {
     leaf config-pre-validation-vpc-device-name {
             type string;
     leaf config-pre-validation-rcm-device-name {
             type string;
     }
  }
  ```

  This config validation of files is also optional. If you do not want to validate the configs, you can turn-off this feature using configurable parameter. If config validation is turned off, then any error in the configuration files results in a config push error, and should be rolled back.

  ```
  container
  configurable-parameters {
     leaf config-pre-validation-required {
             type boolean;
             default false;
     }
     }
    }
  ```

  This config metadata contains all configurable parameters.

Onboarding ESC and Openstack as Devices

For ESC installation, see ESC documentation. Prior to configuration or onboarding and instantiation of VNFs, perform the following setup steps:

NSO and ESC Environment Setup for NFV

1. SSH to ESC host using username and password

   ```
   ssh esc@<esc-ip>
   ```

2. Become Sudo user

   sudo su

3. Edit the following file: `vi /opt/cisco/esc/esc_database/etsi-production.properties`

4. Edit the information as shown below and save the file (Don't change anything in spring user and password). Change the NSO details accordingly. Use only local subnet management IP for communication, and not the floating-IP between ESC/NSO communication.

   ```
   spring.security.user.name=esc
   spring.security.user.password=$1$J7BUBX$Ce4vqA6JcrWCggRpYrPYg1
   ```

```
security.pam.service=
server.additionalConnector.port=8253
server.additionalConnector.key-alias=esc
server.esc.key-alias=esc

nfvo.apiRoot=<NSO-IP>:9191
nfvo.httpScheme=http
nfvo.userName=<NSO-User-name>
nfvo.password=<NSO-Password>
nfvo.authenticationType=BASIC

server.host=<ESC-Orch-IP>
http.enabled=true
https.enabled=false
certificate.validation=false
spring.datasource.password=${PGSQL_PASSWORD}
spring.flyway.password=${PGSQL_PASSWORD}
```

5. Restart the **escadm** service, as shown below:

```
escadm restart
Stopping esc_service: [OK]
Stopping escadm service: [OK]
Starting escadm service: [OK]
#
```

6. Check for the escadm health till it becomes healthy, as shown below (It may take few minutes):

```
escadm health
============== ESC =================
vimmanager (pgid 18651) is running
monitor (pgid 18688) is running
mona (pgid 18741) is running
snmp is disabled at startup
etsi (pgid 19316) is running
pgsql (pgid 18944) is running
portal (pgid 19355) is running
confd (pgid 18978) is running
escmanager (pgid 19131) is running
=====================================
ESC HEALTH PASSED
```

7. Login to the NSO and modify the configs according to the environment and save it into a file:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <nfv xmlns="urn:etsi:nfv:yang:etsi-nfv-descriptors">
  <settings xmlns="http://cisco.com/ns/nso/cfp/cisco-etsi-nfvo">
    <image-server>
      <ip-address><NSO-IP></ip-address>
      <port>8010</port>
      <document-root>/var/opt/ncs/vnfpackages</document-root>
    </image-server>
    <etsi-sol3>
      <server>
        <ip-address><NSO-IP></ip-address>
        <port>9191</port>
        <use-ssl>false</use-ssl>
        <document-root>/var/opt/ncs</document-root>
        <auth-enabled>true</auth-enabled>
        <auth-types>
          <basic>
            <username><NSO-USERNAME></username>
            <password><NSO-PASSWORD></password>
          </basic>
        </auth-types>
```

```
      </server>
      <vnfm-behaviour>
        <vnfm-behaviour-override>
          <id>default-sol3</id>
          <rpc-behaviour>
            <rpc>
              <include>
                <vim-info>false</vim-info>
              </include>
            </rpc>
            <modify>
              <pre>
                <rpc>false</rpc>
              </pre>
              <post>
                <rpc>true</rpc>
              </post>
            </modify>
          </rpc-behaviour>
          <grant>
            <store-history>false</store-history>
            <heal>
              <authorise-grant>true</authorise-grant>
            </heal>
          </grant>
          <onboarding>
            <store-details>true</store-details>
          </onboarding>
        </vnfm-behaviour-override>
      </vnfm-behaviour>
    </etsi-sol3>
  </settings>
  </nfv>
</config>
```

8. Compile all the packages in package folder and perform package reload.

```
ubuntu@test-nso:/var/opt/ncs/packages$ ncs_cli -C
User ubuntu last logged in 2021-09-23T08:00:34.649202+00:00, to test-nso, from
209.165.200.225 using cli-ssh
ubuntu connected from 209.165.200.225 using ssh on test-nso
ubuntu@ncs# packages reload
```

9. Load merge the file, as shown below. This step enables NSO as NFVO and runs NFVO service in 9191 port:

```
ubuntu@test-nso:~$ vi config.xml
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-04T09:10:55.819283+00:00, to test-nso, from
209.165.200.226 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# config
Entering configuration mode terminal
ubuntu@ncs(config)# load merge config.xml
Loading.
1.54 KiB parsed in 0.01 sec (128.38 KiB/sec)
ubuntu@ncs(config)# commit
```

10. Update NACM rule by adding NSO username to “ncsadmin” group

```
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-06T09:56:26.370979+00:00, to test-nso, from
```

```
209.165.200.227 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# config
Entering configuration mode terminal
ubuntu@ncs(config)# nacm groups group ncsadmin user-name ubuntu
ubuntu@ncs(config-group-ncsadmin)# commit
```

11. Copy the necessary packages to the standard location on the NSO (typically /var/opt/ncs/packages).
12. Perform package reload and check for package status. Status should be UP for all packages.

```
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-06T09:58:39.866838+00:00, to test-nso, from
209.165.200.227 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# packages reload
ubuntu@ncs# show packages package oper-status

NAME                  UP  PROGRAM CODE ERROR  JAVA UNINITIALIZED PYTHON UNINITIALIZED
 *
-------------------------------------------------------------------------------
cisco-etsi-nfvo        X      -                    -             -
cisco-rcm-nc-1.0       X      -                    -             -
cisco-staros-cli-5.38  X      -                    -             -
esc                    X      -                    -             -
etsi-sol003-gen-1.13   X      -                    -             -
mobility-common        X      -                    -             -
mop-automation         X      -                    -             -
mop-common             X      -                    -             -
nfv-common             X      -                    -             -
nfv-device-onboarding  X      -                    -             -
nfv-vim                X      -                    -             -
nfv-vnf-lcm            X      -                    -             -
openstack-cos-gen-4.2  X      -                    -             -
```

13. Setup the notification stream: Update /etc/ncs/ncs.conf file to add "nfv-events" stream.

```
<ncs-config>
    <event-streams>
      <notifications>
        <stream>
          <name>nfv-events</name>
         <description>Generic netconf notification stream for NFV events</description>

          <replay-support>true</replay-support>
          <builtin-replay-store>
            <enabled>true</enabled>
            <dir>${NCS_RUN_DIR}/state</dir>
            <max-size>S10M</max-size>
            <max-files>50</max-files>
          </builtin-replay-store>
        </stream>
      </event-streams>
    </notifications>
</ncs-config>
```

14. Restart NSO as sudo user.

```
/etc/init.d/ncs stop
Stopping ncs (via systemctl):                              [  OK  ]
/etc/init.d/ncs start
Starting ncs (via systemctl):                              [  OK  ]
```

15. Onboard NETCONF, ESC, ETSI SOL003 ESC, and Openstack as devices in NSO via device onboarding APIs.

    a. Onboard Openstack as a device. The following is an example. Customize to the specific deployment. This can be configured via NSO CLI in the configuration mode. See NSO documentation for information about authgroup.

    ```
    devices device openstack
    address 209.165.200.228
    port 5000
    authgroup openstack
    device-type generic ned-id openstack-cos-gen-4.2
    ```

    b. Onboard ESC ETSI interface as a device. The following is an example. Customize to the specific deployment.

    ```
    devices device esc-etsi
    address 209.165.200.229
    port 8250
    authgroup esc-etsi
    device-type generic ned-id etsi-sol003-gen-1.13
    ```

    c. Onboard ESC native NETCONF interface as a device. The following is an example. Customize to the specific deployment.

    ```
    devices device esc-netconf
    address 209.165.200.229
    ssh host-key ssh-rsa
    key-data "AAAAB3NzaC1yc2EAAAADAQABAAABAQDYwNCaa3ghJtnJSvn/
    aSPjCuoMKmssZds+J5d9JCOS\n3h3V/fCtJwiH7qMgMXnNc0LEr1fZhxQ4kg5o/
    IafmoYD7N+w/ECqWEp68sjeN+AftiZ9J74D\n+/KDonffgBCHxIVEo0XHYlojrtmpg/
    EH9/N3fQgoSzEhGItGG4uMaAzbWr1pO8AApOPlPi4r\nciL4Qemi6u4i/
    HGFr8MqQp5qcMFd8O30OlB1q1vKn9sq/9sL6EzqyUd2lMounDglEQYMgi8J\
    nyG6upsOFuvhiYRC9qfHML45quyepsJdVi2Li2QwUJLa89EDh148RlhLTJs4s2iAwBGNdvLdK\ntzLu2VGyWKqH"
    !
    authgroup esc-netconf
    device-type netconf ned-id esc
    ```

16. Track the device addition status as shown below (for different devices):

```
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-06T10:09:23.550686+00:00, to test-nso, from
209.165.200.227 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# show vnf-status instances esc-netconf

INSTANCE ID  TIMESTAMP   FUNCTION TYPE OPERATION  STATUS   STATUS MESSAGE
-------------------------------------------------------------------------------
esc-netconf  2021-07-21 *    -          init   success   Device Onboarding initialized

             2021-07-21 *    -          init   success   Device Onboarding initialized

             2021-07-21 *    - fetch-ssh-keys  success   fetch-ssh-keys was successful

             2021-07-21 *    -         connect success   connect was successful
             2021-07-21 *    -       sync-from success   sync-from was successful
             2021-07-21 *    - device-config   success   Subscribed to ESC Netconf
 notification escEvent Stream
             2021-07-21 *    -          ready  success   Device Successfully onboarded
```

Prerequisites for VNF Instantiation

Before submitting the VNF deployment request, make the following configuration changes:

1. **Configurable parameters**

   Set the following configurable parameters, if required:

   - configurable-parameters device-ping-sleeptimesec 30 (default value is 30 sec)
   - configurable-parameters device-ping-retries 150 (default value is 30. In case of RCM, configure it to some higher value, for example, 150)
   - configurable-parameters p2p-required true (default value is false)
   - configurable-parameters p2p-soFile-path /var/opt/ncs/patch_libp2p-2.64.1418.so.tgz

2. **Prepopulating of config metadata**

   When you configure Config-metadata, the device name must be the same as VNF instance name.

   You can call this action from RESTCONF, as shown in the following example:

   URI:
   http://<NSO-IP>:<NSO-REST-PORT>/restconf/data/mobility-common:config-metadata/config-metadata-request

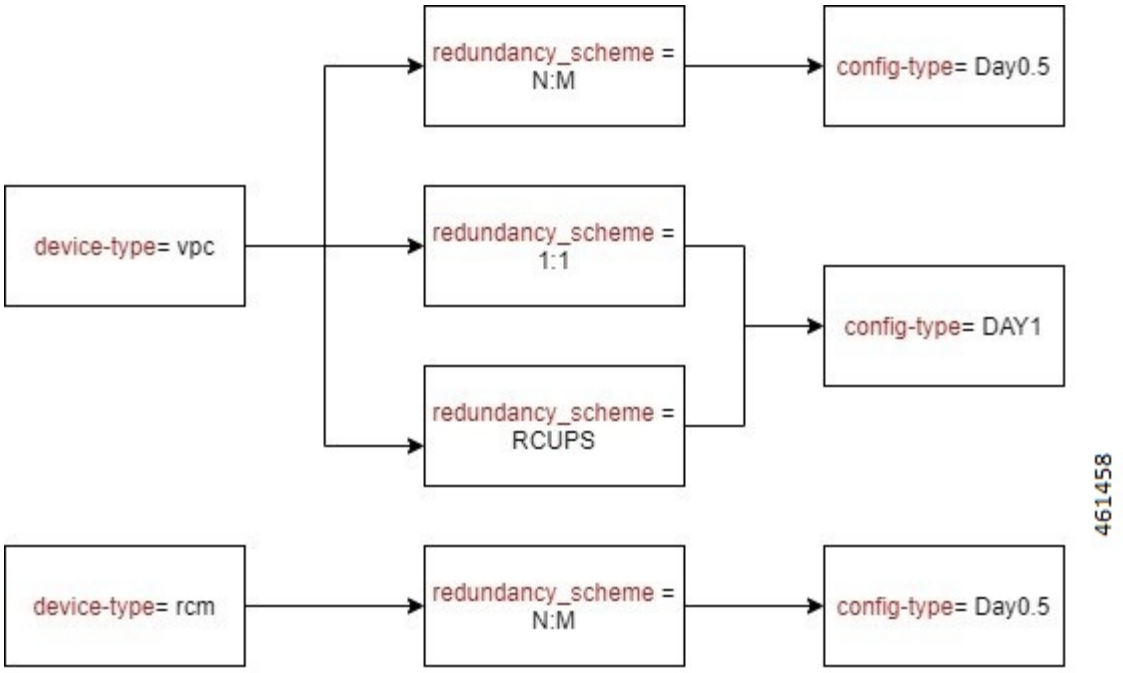
   Method: POST

   Content-Type: application/yang-data+json

   Sample Payload:

```
{
        "config-metadata": {
            "device-name": "test2",
     "schema" : "1:1",
            "attributes":{
                "attribute-name":"test",
                 "attribute-value": "gh"
                  },
            "configuration-type":{
                "config-type": "Day1",
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/test.txt"
                    },
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/day0.5.txt"
                    }

            }
        }
    }
```

Ensure to follow the criteria described in the following figure while pre-populating config metadata:



VNF Instantiation

VNF is instantiated upon configuration. So, to instantiate the VNF, you must load the VNF configuration into the NSO. The VNF has references to the SOL006 VNFD. It also has references to VIM artifacts like Openstack tenant networks, and IP addresses. For details about YANG definition of VNF, see Appendix A: YANG definition of VNF.

Instantiating a VNF involves many components:

- An ETSI SOL001 VNFD template packaged as a TOSCA VNF package
- An ETSI SOL006 VNFD with the same name or ID as the VNF package
- A VNF instance that is proprietary to the NSO

The Mobility function pack ships with some example VNF packages, which also contain the corresponding SOL006 VNFD. These examples can be used as a base, but additional customization is required to fit the deployment. An example VNF configuration is given below:

```
{
   "nfv-vnf-lcm:nfv-vnf":[
      {
         "network-function-type":"VPC-SI",
         "name":"test026",
         "vnfd":"VPC-SI-2P-IMAGE-BOOT",
         "instantiation-level":"default",
         "deployment-flavor":"default",
         "mgmt-user-name":"admin",
         "mgmt-password":"Csco@123",
         "host-name":"vpc-si",
         "domain-name":"cisco.com",
         "ntp-server":"209.165.201.1",
```

```
"name-server":"209.165.201.2",
"location":{
   "vim":{
      "name":"openstack",
      "project":"test",
      "zone-id":"nova"
   },
   "vnfm":"esc-etsi"
},
"network":[
   {
      "type":"VIM_NETWORK_MANAGEMENT",
      "extent":"external",
      "name":"test-mgmt",
      "subnet-name":"test-mgmt-subnet"
   },
   {
      "type":"VIM_NETWORK_ORCHESTRATION",
      "extent":"external",
      "name":"test-orch",
      "subnet-name":"test-orch-subnet"
   },
   {
      "type":"VIM_NETWORK_SERVICE_1",
      "extent":"external",
      "name":"service1",
      "subnet-name":"service1"
   },
   {
      "type":"VIM_NETWORK_SERVICE_2",
      "extent":"external",
      "name":"service2",
      "subnet-name":"service2"
   }
],
"unit":[
   {
      "type":"VPC-SI",
      "image":"core-si-21.23",
      "flavor":"core-si",
      "connection-point":[
         {
            "name":"nic0",
            "ip-address":[
               {
                  "id":0,
                  "fixed-address":[
                     "209.165.201.3"
                  ]
               }
            ],
            "security-group":[
               "default"
            ],
            "network-type":"VIM_NETWORK_ORCHESTRATION"
         },
         {
            "name":"nic1",
            "ip-address":[
               {
                  "id":0,
                  "fixed-address":[
                     "209.165.201.4"
                  ]
```

```
                        }
                     ],
                     "security-group":[
                        "default"
                     ],
                     "network-type":"VIM_NETWORK_MANAGEMENT"
                  },
                  {
                     "name":"nic2",
                     "ip-address":[
                        {
                           "id":0,
                           "fixed-address":[
                              "209.165.201.5"
                           ]
                        }
                     ],
                     "security-group":[
                        "default"
                     ],
                     "network-type":"VIM_NETWORK_SERVICE_1"
                  },
                  {
                     "name":"nic3",
                     "ip-address":[
                        {
                           "id":0,
                           "fixed-address":[
                              "209.165.201.6"
                           ]
                        }
                     ],
                     "security-group":[
                        "default"
                     ],
                     "network-type":"VIM_NETWORK_SERVICE_2"
                  }
               ]
            }
         ],
         "extra-parameters":[
            {
               "name":"BOOTUP_TIME",
               "value":"100"
            },
            {
               "name":"LICENSE_KEY",
               "value":"\"VER=1|DOI=1624646484|DOE=1640457684|ISS=3|NUM=212017|
CMT=SWIFT_License|LSG=5000000|LEC=10000000|LGT=5000000|FIS=Y|FR4=Y|FTC=Y|FSR=Y|
FPM=Y|FID=Y|FI6=Y|FLI=Y|FFA=Y|FCA=Y|FTP=Y|FTA=Y|FDR=Y|FDC=Y|FGR=Y|FAA=Y|FDQ=Y|
FEL=Y|BEP=Y|FAI=Y|FCP=Y|LCF=5000000|LPP=5000000|LSF=5000000|FLS=Y|FSG=Y|
LGW=5000000|HIL=XT2|LSB=5000000|LMM=5000000|FIB=Y|FND=Y|FAP=Y|FRE=Y|FHE=Y|
FUO=Y|FUR=Y|FOP=Y|FRB=Y|FCF=Y|FVO=Y|FST=Y|FSI=Y|FRV=Y|F6D=Y|F13=Y|FIM=Y|
FLP=Y|FSE=Y|FMF=Y|FEE=Y|FHH=Y|FIT=Y|FSB=Y|FDS=Y|LSE=5000000|FLR=Y|FLG=Y|
FMC=Y|FOC=Y|FOS=Y|FIR=Y|FNE=Y|FGD=Y|LIP=5000000|FOE=Y|FAU=Y|FEG=Y|FL2=Y|
FSH=Y|FLF=Y|FSP=Y|FNI=Y|FCI=Y|FME=Y|FCN=Y|FUB=Y|FSF=Y|FGO=Y|FPE=Y|FWI=Y|
FAC=Y|FIE=Y|FSM=Y|FAG=Y|FNQ=Y|FEW=Y|FAR=Y|FOX=Y|FPW=Y|FAM=Y|FGX=Y|FWT=Y|
FUA=Y|LDT=5000000|LEX=5000000|LVL=5000000|LQP=5000000|LMP=5000000|
LCU=10000000|LUU=10000000|FXS=Y|FLC=Y|FRT=Y|FSX=Y|FBS=Y|FRD=Y|FXM=Y|
LTO=10000000|FNS=Y|LNS=5000000|SIG=MC0CFBge/
0TZha2Ta7c1L5CLOL2tgDIDAhUAhIKwZxXEJJpr9Xk5buNyzZStrNM\""
            }
         ]
      }
```

```
    ]
}
```

The following is another example to instantiate RCM VNF:

```
{
    "nfv-vnf-lcm:nfv-vnf": [
      {
        "network-function-type": "RCM",
        "name": "RCM-ahhashem-sol003-78",
        "vnfd": "RCM-IMAGE-BOOT",
        "instantiation-level": "default",
        "deployment-flavor": "default",
        "mgmt-user-name": "luser",
        "mgmt-password": "$8$40/jVMTHJY+Jrd7mZiwqdrKEIz6Kc5Pt2Qvnwi0/65g=;”
        "host-name": "rcm",
        "domain-name": "cisco.com",
        "ntp-server": "209.165.201.1",
        "name-server": "209.165.201.1",
        "location": {
          "vim": {
            "name": "openstack",
            "project": "ahhashem",
            "zone-id": "nova"
          },
          "vnfm": "esc-etsi"
        },
        "network": [
          {
            "type": "VIM_NETWORK_MANAGEMENT",
            "name": "ahhashem-mgmt",
            "extent": "external",
            "subnet-name": "ahhashem-mgmt-subnet"
          },
          {
            "type": "VIM_NETWORK_ORCHESTRATION",
            "name": "ahhashem-orch",
            "extent": "external",
            "subnet-name": "ahhashem-orch-subnet"
          },
          {
            "type": "VIM_NETWORK_SERVICE_1",
            "name": "service1",
            "extent": "external",
            "subnet-name": "service1"
          } ,
          {
            "type": "VIM_NETWORK_SERVICE_2",
            "name": "service2",
            "extent": "external",
            "subnet-name": "service2"
          }
        ],
        "unit": [
          {
            "type": "RCM",
            "image": "core-rcm-21.23",
            "flavor": "mkal-rcm-hugepages",
            "connection-point": [
              {
                "name": "nic0",
                "ip-address":{
                    "id": 1,
                    "fixed-address": ["209.165.201.7"]
                },
```

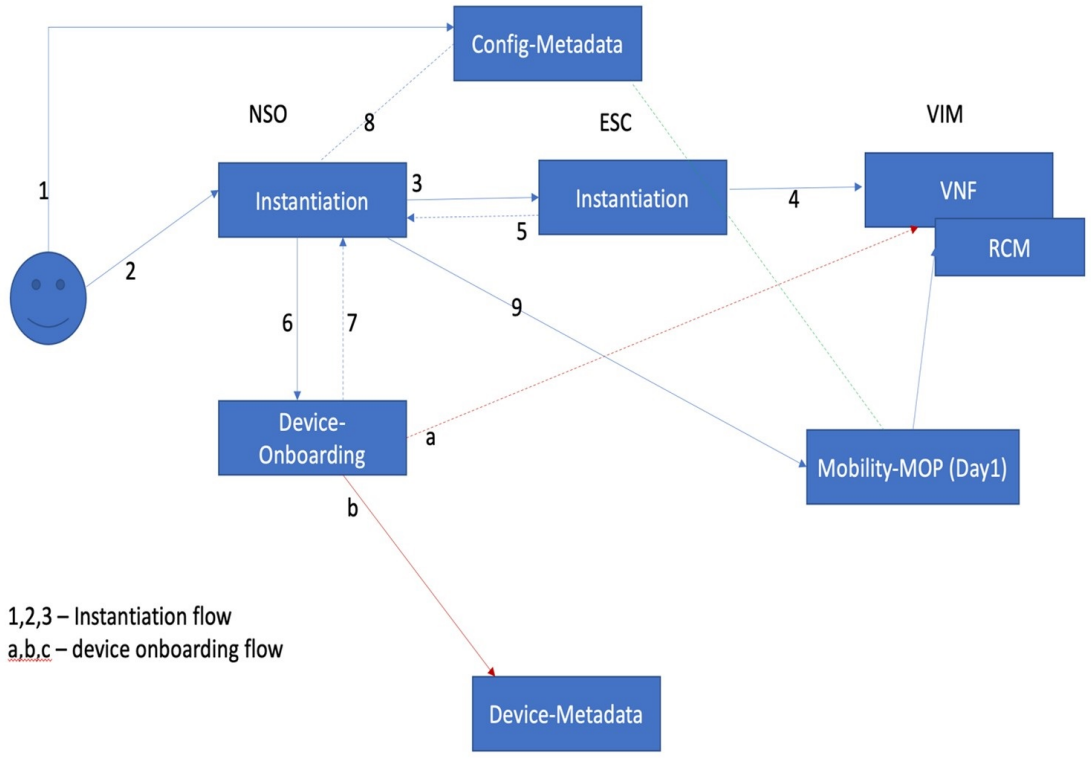
```
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_ORCHESTRATION"
        },
        {
          "name": "nic1",
          "ip-address":{
              "id": 1,
              "fixed-address":["209.165.201.8"]
          } ,
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_MANAGEMENT"
        },
        {
          "name": "nic2",
          "ip-address": {
              "id": 1,
              "fixed-address": ["209.165.201.9"]
          },
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_SERVICE_1"
        } ,
        {
          "name": "nic3",
          "ip-address": {
              "id": 1,
              "fixed-address": ["209.165.201.10"]
          },
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_SERVICE_2"
        }

      ]
    }
  ],
  "extra-parameters": [
    {
      "name": "VIM_VM_NAME",
      "value": "RCM-ahhashem-sol003-78"
    },
    {
      "name": "HOST_NAME",
      "value": "rcm"
    },
      {
        "name": "NIC0_TYPE",
        "value": "virtual"
      },
      {
        "name": "NIC1_TYPE",
        "value": "virtual"
      },
      {
        "name": "NIC2_TYPE",
        "value": "direct"
      },
      {
        "name": "NIC3_TYPE",
        "value": "direct"
      },
      {
        "name": "MGMT_USER_NAME",
        "value": "luser"
      },
      {
```

```
                "name": "MGMT_PASSWORD_ROUND4096",
                "value": "$6$rounds=4096$P2wdTbEBO0LHmHi$OwbVEIarMbt
Qxbu5Us5kW0n0MOWp3QN9eVRX7WjvLm4xTJvFpl6vHez3XkKm39XJJ7dGRRIsZqXfcZRjQBA7E."
            },
            {
                "name": "SERVICE_INTERFACE_IP_1",
                "value": "209.165.201.9"
            },
            {
                "name": "SERVICE_INTERFACE_IP_2",
                "value": "209.165.201.11"
            },
            {
              "name": "NTP_SERVER",
              "value": ["209.165.201.12","209.165.201.13","209.165.201.14"]
            }

        ]
      }
    ]
  }
```

VNF Instantiation - Component Interactions and Flows

The following figure illustrates the complete flow of end-to-end instantiation automation:

Figure 36: VNF Instantiation Interactions



Detailed Steps:

1. The Network operator has all the required details for VNF instantiation including the name, type, dynamic attributes, and the configuration files. The network operator places the config files into the NSO filesystem, and registers the details with NSO config DB for automation.

   This step includes the following tasks:

   - The network operator Secure copy (SCP) the config files into NSO filesystem. This location must be an NFS, or it's replicated in NSO HA environment.
   - Registers all attribute value pairs, dynamic substitution values, Day-0.5, or Day-1 configurations.
   - Enables the validation of config files, and provides the testing device details.
   - If the revalidation flag is set to true, config metadata action internally validates all config files. Otherwise, it fails while applying the configuration.

2. The Network operator prepares the payload for VNF instantiation with all the details. Then, the network operator invokes the payload to create an instance. It does the basic validation and processes the order.

   This step includes the following task:

   - Validates the inputs such as password length, image, flavor, and network existence in Openstack before invoking an order

3. NSO processes the order internally and prepares the ESC VNF instantiation order.

   This step includes the following tasks:

   - Creates NSO footprint of the service
   - Does CSAR validation
   - Invokes ESC VNF instantiation order using SOL3/SOL4 input
   - Starts listening to ESC notifications (both ETSI and NETCONF)

4. ESC performs input validation of SOL3/SOL4 and creates the order in VIM.

   This step includes the following tasks:

   - ESC invokes VNF instantiation.
   - On Successful invocation of VNF, it creates mono monitors to monitor the VNF.
   - Returns the updates via ETSI and NETCONF notifications to NSO (both Success and Failure).

5. ESC returns the periodic updates on progress to NSO via ETSI or NETCONF notifications.

   This step includes the following tasks:

   - ESC constantly sends the ETSI and NETCONF notifications on progress.
   - ETSI notification comprises deploy – init, processing, and completed notification.
   - NETCONF notifications provide more granular information on VM status.
   - On Failure, it gives appropriate error message.

6. On receiving VNF instantiation completion message from ESC, NSO onboards as an NSO device.

This step includes the following tasks:

- Instantiation logic fetches the details from input payload, and invokes device onboarding logic.
- NSO performs the fetch-ssh-host-key from the device.
- NSO performs connection check.
- NSO performs sync-from.
- NSO executes post check command such as “show version” on device.
- NSO adds the device into NSO device tree.

7. NSO instantiation logic waits for device addition to complete.

   This step includes the following tasks:

   - NSO checks if device onboarding process is complete.
   - If device onboarding fails, NSO stops the execution.

8. NSO instantiation logic reads the prepopulated config metadata to interpret the config to be pushed.

   This step includes the following tasks:

   - Reads the prepopulated config metadata and interprets the Day-0.5 or Day-1 configuration files based on the device-name (Device name is based on VNF name)
   - For RCM-based N:M scheme, Day-0.5 is pushed.
   - On 1:1 case, Day-1 is pushed.
   - On missing information, instantiation completes and stops processing.

9. NSO takes the config files from config metadata, formulates the Mobility MOP input format, and invokes the MOP for config push.

   This step includes the following tasks:

   - Invokes the Mobility MOP and gets the task-id.
   - Periodically checks for the status on task-id.
   - Saves the config permanently in device flash if its 1:1 CP or UP pairs (via MOP).
   - Completion status is updated in vnf-status ledger.

Checking the VNF Instantiation Status

You can check the status of VNF instantiation using **vnf-status** command periodically.

Any failure, processing, or completion related messages are appended in the status message.

```
show vnf-status instances vnf-instance-name
INSTANCE ID TIMESTAMP TYPE OPERATION STATUS STATUS MESSAGE
--------------------------------------------------------------------------
<VNF-Name> <Time-Stamp> <type> <function> <status> <message-if-any>
```

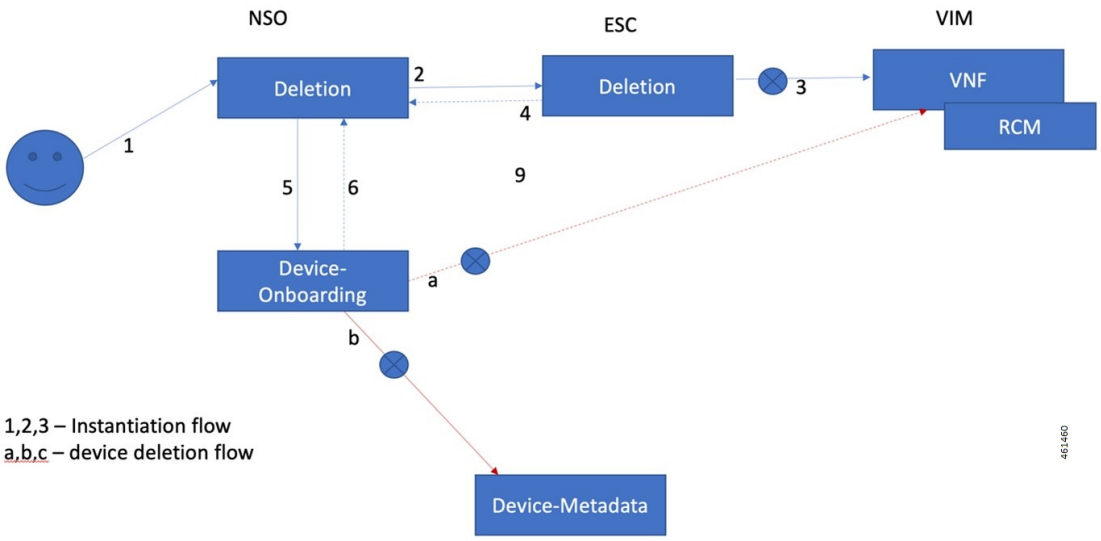
VNF Dashboard

VNF instantiation steps and current status of the VNF are displayed in NSO based dashboard.

VNF Deletion

The following flow diagram illustrates the complete flow of end-to-end deletion automation.

Figure 37: VNF Deletion Interactions



Detailed Steps:

1. Network operator decides to decommission or delete the existing instance, which is in running or failed state.

   This step includes the following tasks:

   - Network operator provides the VNF-name with type for the deletion
   - NSO does the validation of the VNF existence

2. NSO checks for the VNF instance status, and if there is a failed instance at NSO end, NSO invokes ESC for deletion from VIM or perform rollback.

   This step includes the following tasks:

   - NSO decides to push it to ESC or perform rollback (in case of failed instance within NSO)
   - NSO does the asynchronous request to ESC and waits for notifications.

3. ESC does the clean-up and removes the VNF monitors.
4. ESC generates ETSI/NETCONF notifications to NSO.
5. NSO processes ESC notifications and performs the following:

- Invokes the device-onboarding package for deletion of the instance
- Removes entry from “nfv-vnf-inventory”

6. Device onboarding package deletes the device from NSO, and the status is updated in the VNF ledger.

Checking the VNF Deletion Status

You can check the status of VNF deletion using **vnf-status** command.

Any failure, processing, or completion related messages are appended in the status message

```
show vnf-status instances vnf-instance-name
INSTANCE ID TIMESTAMP TYPE OPERATION STATUS STATUS MESSAGE
------------------------------------------------------------------------
<VNF-Name> <Time-Stamp> <type> <function> <status> <message-if-any>
```

Removing Configuration Metadata

This is a manual step, and you need to remove the config metadata using NSO action. Keeping this data doesn’t have any impact.

Cleaning Config Files from NSO Filesystem

You need to remove the config files manually from NSO filesystem. Keeping this data doesn’t have any impact.

Automation Process - VNF Deployment, Onboarding, and Configuration Push

The Automation process includes the following sections:

Instantiation of VNF using Input Payload

After making necessary changes, submit the instantiation request using input payload. The automation process for VNF instantiation starts.

For input payload sample, see the section VNF Instantiation.

Onboarding VNF as a Device in NSO

Upon successful instantiation, the VNF is onboarded as a device on the NSO. The device name is the same as the VNF name.

Upgrading the P2P Module in VPC Device

If the “device-type” is of VPC type and the configurable-parameters “p2p-required” is set to “true” with the “p2p-soFile-path” defined, copy the P2P file to device flash directory, and then upgrade the P2P module.

The P2P module is installed on the device.

Configuration Push to the Onboarded Device

The following are the static parameters that are used during automated config push:

- operation-type =Commit

- mop-type=Common
- save-config-permanently= default is false, and it is set to true when the device type is "vpc"

Once the config push is done using the configuration files, a task-id is generated. Using the task-id, it checks the status of config-push, and based on the status, the ledger entry is updated.

Note NSO doesn't perform configuration audit on RCM. If an RCM reboots when NSO is in the process of pushing configuration to it, the NSO doesn't re-push the configuration upon reboot completion. The configuration must be re-pushed manually. NSO alerts the operator about configuration push failure. Any configuration successfully pushed to the RCM is persistent across reboots of that RCM.

Appendix A: YANG definition of VNF

This section provides a sample YANG definition of VNF.

```
module nfv-vnf-lcm {
  namespace "http://com/cisco/cx/servicepack/nfv/vnflcm";
  prefix nfv-vnf-lcm;

  import ietf-inet-types { prefix inet; }
  import tailf-common { prefix tailf; }
  import tailf-ncs { prefix ncs; }
  import nfv-common { prefix nfv-common; }
  import tailf-kicker { prefix kicker; }
  include nfv-vnf-lcm-nano {
    revision-date 2020-02-14;
  }

  organization "Cisco-AS";

  contact "Cisco AS";

  description "Generic NFV VNF LCM service package";

  revision 2020-10-22 {
    description "Active Inventory and LCM Auto/on-demand heal support";
  }

  revision 2020-07-01 {
    description "Re-branded per new naming convention";
  }

  revision 2020-02-14 {
    description "First version, ready for testing";
  }

  notification vnf-lcm {
    description "Notification about Network Function Operation";
    uses nfv-common:network-function-notification;
  }

  notification vnf-alarm {
    description "VNF alarms";
    uses nfv-common:vnf-alarm;
  }
```

```
container nfv-vnf-inventory {
  tailf:info "CDB model to persist the VNFs, associated project, VIM and the
      VM details";
  config false;
  tailf:cdb-oper {
    tailf:persistent true;
  }

  list vnf {
    tailf:info "VNFs with associated VMs and status";
    key name;
    leaf name {
      tailf:info "VNF Name";
      type string;
    }
    leaf vnfd {
      type string;
      tailf:info "Associated VNFD name";
    }
    leaf project {
      type string;
      tailf:info "Associated vim tenant/project";
    }
    leaf vim {
      type string;
      tailf:info "Associated VIM";
    }
    leaf status {
      type string;
      tailf:info "Overall VNF status";
    }
    list vm {
      tailf:info "Associated VMs and the status";
      key name;
      leaf name {
        type string;
        tailf:info "VM name";
      }
      leaf type {
        type string;
        tailf:info "VM Type";
      }
      leaf flavor {
        type string;
        tailf:info "VIM flavor that is used to deploy the VM";
      }
      leaf host {
        type string;
        tailf:info "Compute host where the VM has been deployed";
      }
      list connection-point {
        key nic-id;
        leaf nic-id {
          type uint8;
          tailf:info "NIC id of the connection point";
        }
        leaf ip-address {
          type inet:ip-address;
          tailf:info "IP address of the connection point";
        }
      }
      leaf status {
        type string;
        tailf:info "VM status";
```

```
      }
    }

    leaf netconf-notification-done {
        tailf:hidden nfv-internal;
        type empty;
    }
  }
}

list nfv-vnf {
  description "Generic RFS model for VNF LCM";

  key "network-function-type name";

  leaf network-function-type {
    tailf:info "virtual network function type";
    type enumeration {
      enum "VPC-SI";
      enum "VPC-DI";
      enum "CSR1KV";
      enum "GENERIC";
      enum "VCU";
      enum "VDU";
      enum "EMS";
      enum "RCM";
    }
  }

  leaf name {
    tailf:info "Unique service id";
    type string;
  }

  leaf vnfd {
    mandatory true;
    type string;
    tailf:info "VNFD to use for this type of Network Function that has to be
          onboarded on the target VIM.";
  }

  uses ncs:service-data;
  ncs:servicepoint nfv-vnf-lcm;
  uses ncs:nano-plan-data;

  tailf:action heal {
    tailf:info "Heal VNF";
    tailf:actionpoint nfv-lcm-heal-ap;
    input {
    }
    output {
      uses nfv-common:standard-action-response;
    }
  }

tailf:action start {
          tailf:info "Start VNF";
          tailf:actionpoint nfv-lcm-start-ap;
          input {
          }
          output {
              uses nfv-common:standard-action-response;
          }
      }
```

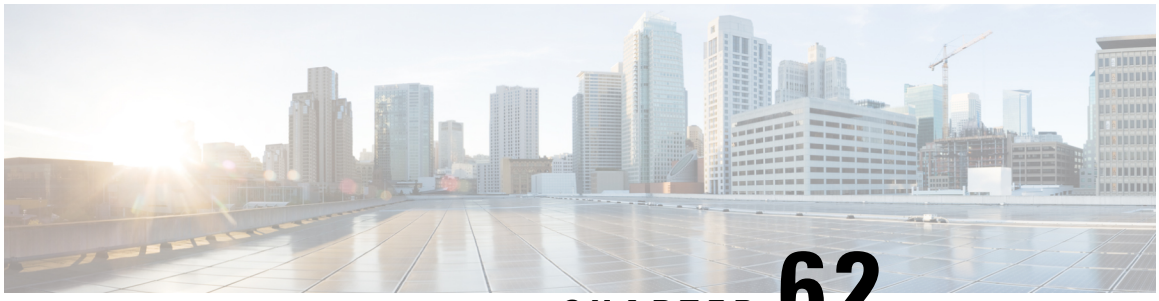
```
 tailf:action stop {
           tailf:info "Stop VNF";
           tailf:actionpoint nfv-lcm-stop-ap;
           input {
           }
           output {
               uses nfv-common:standard-action-response;
           }
       }

tailf:action scale {
           tailf:info "Scale-In VNF";
           tailf:actionpoint nfv-lcm-scale-ap;
           input {
               leaf scale-type {
                   mandatory true;
                   tailf:info "SCALE IN or OUT";
                   type enumeration {
                       enum "OUT";
                       enum "IN";
                   }
               }

               leaf no-of-instances {
                   tailf:info "Number of scale IN or OUT instances. Default is 1";
                   type uint32;
                   default 1;
               }

               leaf vdu-type {
                   mandatory true;
                   tailf:info "vdu-type as CF/SF/VPC-SI etc";
                   type string;
               }

           }
           output {
               uses nfv-common:standard-action-response;
           }
       }

     tailf:action retry {
           tailf:info "Stop VNF";
           tailf:actionpoint nfv-lcm-retry-ap;
           input {
           }
           output {
               uses nfv-common:standard-action-response;
           }
       }

   leaf instantiation-level {
     type string;
     default "default";
     tailf:info "Instantiation level defined in VNFD to use. This will determine
           the number of VMs/VDUs to be deployed.";
   }

   leaf deployment-flavor {
     type string;
     default "default";
     tailf:info "Deployment flavor defined in the VNFD to use. Describes a specific
           deployment version of a VNF with specific requirements for capacity
```

```
        and performance.";
}
leaf mgmt-user-name {
  type nfv-common:identifier;
  description " Management login username specific to this VNF. Default values
        can be configured per VNF type.";
}

leaf mgmt-password {
  tailf:suppress-echo "true";
  type tailf:aes-cfb-128-encrypted-string;
  description "Management login password specific to this VNF.";
}

leaf host-name {
  type inet:domain-name;
  description "Hostname to use to communicate with this network function";
}

leaf domain-name {
  type inet:domain-name;
  description "Domain name used to construct Fully Qualified Domain Name by
        concatenating with VM hostname: <hostname>.<domain>";
}

leaf ntp-server {
  description "NTP server to use for VNFs deployed in this data center";
  type inet:host;
}

leaf name-server {
  type inet:ip-address;
  description "Name server";
}

container location {
  container vim {
    leaf name {
      description "NFVI this Network Function is deployed on.";
      type leafref {
        path "/ncs:devices/ncs:device/ncs:name";
      }
      //must "/ncs:devices/ncs:device[ncs:name=current()]/ncs:platform/ncs:name
      //          = 'Openstack'" {
      //  error-message "Please select Openstack devices only";
      //}
    }
    leaf project {
      type nfv-common:identifier;
      description "VIM project used to instantiate VNFs";
      mandatory true;
    }
    leaf zone-id {
      type string;
      default "nova";
      description "VIM zone id";
    }
    //TODO might need to support user domain and project domain
  }
  leaf vnfm {
    mandatory true;
    type leafref {
      path "/ncs:devices/ncs:device/ncs:name";
    }
```

```
    //must "/ncs:devices/ncs:device[ncs:name=current()]/ncs:platform/ncs:name
    //        = 'ETSI SOL'" {
    //  error-message "Please select ETSI-SOL VNFM devices only";
    //}
    description "ESC VNFM onboarded";
  }
}
list network {
  key type;
  leaf type {
    type nfv-common:identifier;
  }
  leaf name {
    type nfv-common:identifier;
    mandatory true;
  }
  leaf extent {
    type nfv-common:network-extent;
  }
  leaf subnet-name {
    when "../extent='external'";
    type nfv-common:identifier;
    mandatory true;
  }
}

list unit {
  description "Virtual Deployment Unit, a single VM.";
  key type;

  leaf type {
    description "VDU type as defined in the VNFD of this Network Function.";
    type nfv-common:identifier;
  }
  leaf image {
    type string;
    description " Image to use for this type of Network Function. Must have been
            be onboarded on the target VIM.";
  }
  leaf flavor {
    mandatory true;
    type string;
    description " Flavor to use for this type of Network Function. Must have been
            onboarded on the target VIM.";
  }
  list storage-volume {
    key id;
    description "Out of band Storage volumes to use for this network function";
    leaf id {
      type string;
    }
    leaf volume-name {
      type string;
      description "Storage Volume to use for this type of Network function";
    }
  }
  list connection-point {
    key name;
    description " Network connection point such as a network interface card, as
            defined in the descriptor.";
    leaf name {
      mandatory true;
      type nfv-common:identifier;
    }
```

```
        list ip-address {
            key id;
            ordered-by user;
            leaf id {
             type uint8;
                tailf:info "IP Address ID for connection points";
            }
            leaf-list fixed-address {
                ordered-by user;
                description " IP address(es) to assign this network interface for both
scaled and non-scaled VNF's. Both IPv4 and
                        IPv6 is possible to allow for dual-stack cases if this VNF requires

                          it for Internet access.";
                type inet:ip-address;
             }
        }

        list vip {
          key address;
          ordered-by user;
          description " Virtual IP address(es) to assign this network interface. Both
                    IPv4 and IPv6 is possible to allow for dual-stack cases if this
                    VNF requires it for Internet access. Setting this will populate
                    allowed-address-pair list in the CVIM";

          leaf address {
            type inet:ip-address;
          }
          leaf netmask {
            type inet:ip-address;
            mandatory true;
          }
        }
        leaf-list security-group {
          type nfv-common:identifier;
          description "Security group(s) to apply to this network interface.";
        }
        leaf network-type {
          type leafref {
            path "../../../network/type";
          }
          description "Network used for this connection-point.";
        }
      }
    }
    list extra-parameters {
      description "VNF instance specific additional parameters defined in the VNFD.
            This will override the values configured in the VNFD";
      key name;
      leaf name {
        type string {
          pattern "[A-Za-z0-9_]+";
        }
      }
      leaf value {
        type string;
      }
    }
  }

   list nfv-retry-vnfs {
        tailf:info "Retry VNF's to tweak the notifications";
```

```
            config false;
            tailf:cdb-oper {
                tailf:persistent true;
            }
            tailf:hidden nfv-internal;

            key name;
            leaf name {
                tailf:info "VNF Name";
                type string;
            }
        }
    }
```



CHAPTER 62

NSH Traffic Steering

- Revision History, on page 577
- Feature Description, on page 577
- How it Works—Standalone Mode, on page 582
- Configuring the L2 and NSH Traffic Steering Feature—Standalone Mode, on page 586
- Monitoring and Troubleshooting—Standalone Mode, on page 595
- Feature Description—Sandwich Mode, on page 602
- How it Works—Sandwich Mode, on page 604
- Configuring NSH Traffic Steering—Sandwich Mode, on page 609
- Configuring Post Processing Ruledef in Both Standalone and Sandwich Mode, on page 612
- Configuring BFD Instance Id Using Interface Name in UP Appliance Group, on page 612
- Monitoring and Troubleshooting the NSH Traffic Steering—Sandwich Mode, on page 613

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
The support for post processing rule condition match for Traffic Steering and L2 up-appliance-group BFD configuration is available in this release.	21.23.22
First introduced.	Pre 21.24

Feature Description

The 3GPP EPC architecture enables data traffic steering across various service functions on the Gi interface. The traffic steering architecture is based on the Network Service Header (NSH) service chaining protocol. The EPC gateway needs to perform the traffic steering to steer the traffic across the multiple service chains containing the appliances which support NSH.

The following are the two modes of NSH Traffic Steering:

- Standalone Mode
- Sandwich Mode

This feature enables the charging and steering of traffic to be independent of each other based on the customer's requirement. It is possible for customers to include a large set of traffic categories for steering traffic with minimum configurational enhancements within the existing use case scenarios.

Post Processing Rule Condition Match for Traffic Steering

A simple traffic classification helps in simplifying the operation and configuration processes in traffic steering due to the huge number of the charging rules across multiple rulebases.

- Trigger condition in service scheme framework supports post processing ruledef name match.
- The L3/L4 ruledef which is configured as a post processing rule for traffic is traffic-steered.
- Trigger action supports trigger condition of post processing rule match for traffic steering.
- The post processing ruledef name in trigger condition is supported in PFD push and RCM.

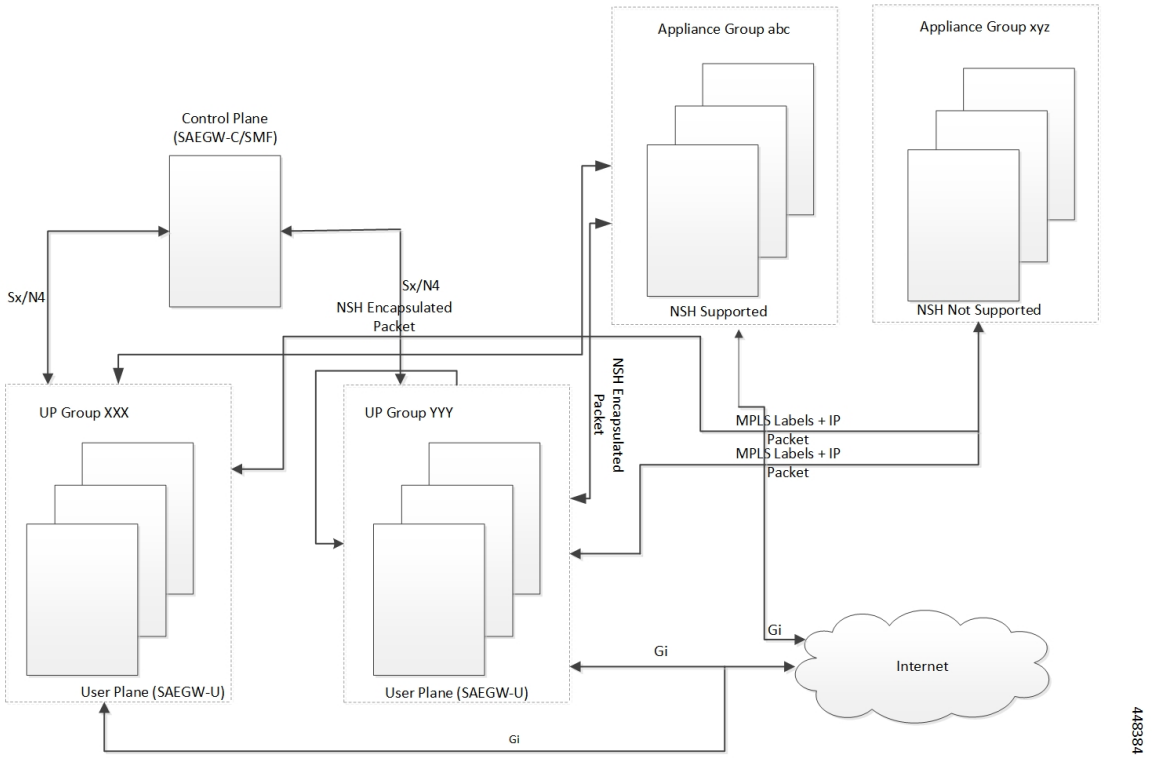
BFD Instance Id Configuration in UP Appliance Group Using Interface Names

For traffic steering, the configuration of Bidirectional Forwarding Detection (BFD) instance id in the **up-appliance-group** is enabled using interface names along with IP configuration.

Architecture—Standalone Mode

The following figure illustrates the architectural setup for CUPS based gateway for NSH appliances.

Figure 38: NSH Traffic Steering Architecture—Standalone Mode



The feature supports a service function chain for NSH supported appliances. The gateway is configured to select a suitable steering or encapsulation method for steering traffic that is based on each appliance instance or group.

Table 42: Call Flow

Step	Description
1.	UL packet received at the SAEGW-U is classified based on the configured policy associated with the appropriate SFC.
2.	The Saegw performs the SFP selection based on the stickiness (MSISDN stickiness) or service and load availability of the SFPs. The UL traffic is NSH (IP-UDP) encapsulated steered on the selected SFP with the context header populated as necessary.
3.	The NSH appliance on receiving the NSH Packet, processes the IP packet (and possibly the context header) and sends the packet over the Gi interface.
4.	Destination server sends the DL packet from the Gi interface to the SAEGW-U. The DL traffic is NSH (IP-UDP) encapsulated steered on the selected SFP with the context header populated as necessary.

Step	Description
5.	The NSH appliance on receiving the NSH Packet, processes the IP packet (and possibly the context header) and hairpin the packet back to the SAEGW-U.
6.	The SAEGW-U on receiving the NSH packet: • Decapsulates the received payload • Processes the IP packet (and possibly the context header) and send the packet over the Gn interface to the UE

Components

The traffic steering architecture comprises of the following main components:

Control Plane (SAEGW-C)

CP sends information to UP on how to steer the subscriber's traffic. The UP steers all or only part of the subscriber data traffic that is based on policies that are defined for the subscriber. It's possible to steer different types of subscriber traffic to different service function chains.

CP selects the service chain name for a subscriber after it receives the Ts-subscription-scheme AVP from PCRF, which is based on locally configured policies.

User Plane (SAEGW-U)

Based on the policy, which UP receives from CP, it steers the subscriber data traffic to one or more service function chains.

UP also performs the following functions:

- Select a Service Function Path (SFP) for a particular Service Function Chain (SFC).
- Maintain subscriber stickiness while forwarding traffic toward the appliances.
- If a node or an appliance fails, reselect and steer the subscriber data traffic to a new node.
- Manage **In-Service** and **Out-of-Service** status for SFPs.
- Manage SFC status depending on the number of serviceable SFPs available within an SFC.

NSH

For monitoring health of the NSH appliance, each SAEGW-U/UPF is responsible for monitoring of the appliance load and serviceability stat.

- Use the OAM NSH packet mechanism to monitor the status of the appliances.
- The monitoring frequency for the configuration is (1-20 secs) with a default interval of 1 sec.
- In case the OAM request times out. Do the retry. The timeout and the retry value are configurable with values of 1-5 secs for timeout (default of 3 secs) and 1-3 retries (default of two retries).

- In addition to the appliance serviceability status, the current load on the appliance is under observation. Monitor the current load in order to maintain the optimum load balancing among various instances of an SF. This load status returns through the NSHs OAM response packet.

Limitations

The NSH traffic steering has the following limitations:

- On NSH appliance, make sure that the interface fragmentation doesn't happen. Keep the MTU towards the NSH appliance interface bigger than Gn/Gi interface.
- For HTTP pipelined sessions, mid flow HTTP partial packets, and TCP Out of order packets, if requires an SFP revaluation with L7 conditions, doesn't reach the NSH appliance.
- If you remove the SFP ID configuration from the main configuration, show configuration still shows the SFP ID. The SFP ID goes away once committed to VPP, using the commit CLI.
- Traffic steering statistics indicate the packets which are candidate for traffic steering. In traffic steering statistics those packets are also counted which are dropped by quota exhaust, though they still are the candidates for traffic steering.
- If modification of NSH SRC/bind IP address OR appliance IP address is required in the configuration for any NSH appliance's instance, then you need to remove the instance, then SFPs associated with it, put the SFPs and new instance with modified IP addresses. Perform the commit afterwards.
- When node failure is done and continuous data is coming, then there can be discrepancy in steering statistics. Data steered on SFPs which is going down is not reflected in statistics.
- For multi PDN call, NSH instance stickiness is restricted to each subscriber session.
- In case of a change in the state at the SAEGW-U due to ICSR or config change like SFP removal in the interim period, there is a possibility that packets which are being hair pinned back from the appliance in this window can be dropped. All further incoming packets are processed as normal
- In case of the first packet of a flow being a DL packet (session recovery), just that first packet is dropped. However, the retransmitted packet and all subsequent packets are sent out as normal.
- In case of change in the NSH format tags, tag types stream-fp-md encode, reverse-stream-fp-md , secondary-srv-path-hdr, and rating-group comes into effect for new flows and not for existing flows. Any changes for the remaining tags in the NSH format applies for new sessions while traffic on existing sessions continue with older format tags. In such cases, particularly in case of modification and deletion of tags, the appliance can mismatch the tag values received in the NSH packets and can lead to ambiguous behavior. So, perform the NSH-format type changes carefully.
- Server initiated TCP Flows are not considered for Traffic steering.
- Monsub support for capturing NSH traffic is not currently available.
- For addressing any appliance level limitation (example - traffic type), policy selection configuration on the service scheme provides the flexibility to filter out such traffic from selecting a service chain containing such appliance.
- For N:M setup, service scheme config (trigger action, trigger condition, service-scheme, subscriber class, and subscriber base) needs to be configured in Day-0 config on UP. Service scheme when configured, in common config on UP, is hitting a race condition leading to service scheme not getting configured on user-plane sessmgrs intermittently, which leads to failure of traffic steering functionality.

- OAM stats for L2 steering is partially supported.
- For HTTP concatenated packet, the packet is traffic steered based on the policy matched by the last HTTP GET in the packet.
- In case a appliance goes down, the flow gets onloaded for revaluation when the next uplink packet is recovered on the flow. Post which the a new SFP selection happens and the traffic is steered to the new appliance.

How it Works—Standalone Mode

Packet Flows

This section describes the packet flows for the NSH traffic steering architecture.

Uplink Packets

Figure 39: Uplink Packet Flow

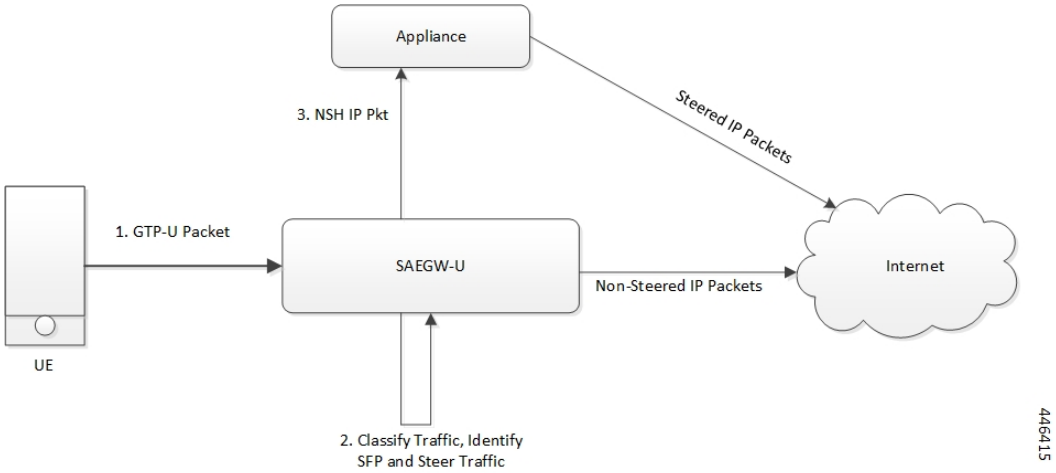


Table 43: Uplink Packet Flow Description

Steps	Description
1	UE sends the subscriber data packets to SAEGW-U.
2	SAEGW-U classifies the subscriber data traffic that is based on subscriber policies, and identifies an SFC to select an SFP accordingly.
3	SAEGW-U steers the Uplink (UL) packets with NSH encapsulation as per NSH RFC and sends to NSH appliance. SAEGW-U sends the non-steered IP packets to the server.
4	NSH supported appliance on receiving uplink packet, takes the decision to forward the packet to server based on certain criteria.

Downlink Packets

Figure 40: Downlink Packet Flow

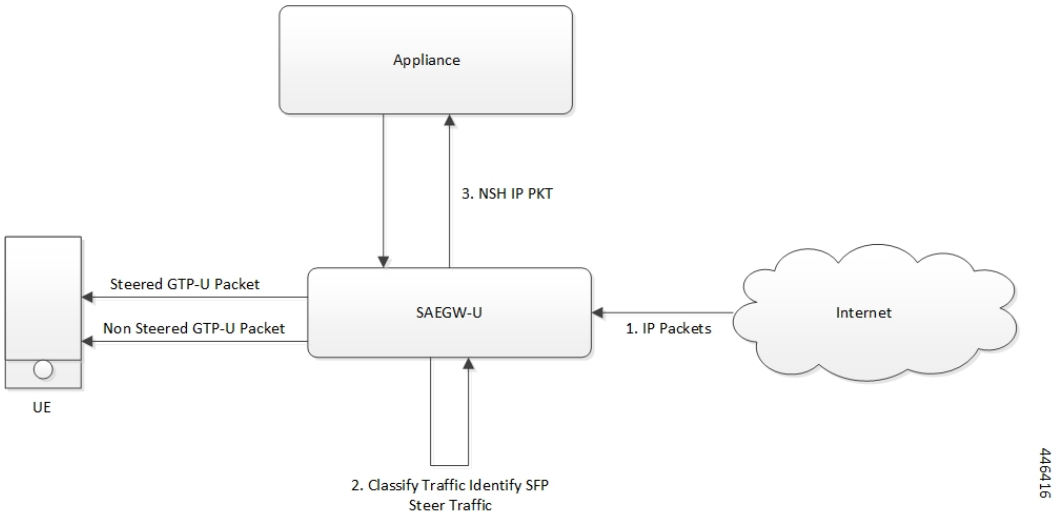


Table 44: Downlink Packet Flow Description

Steps	Description
1	SAEGW-U receives the Downlink (DL) packets from the server.
2	SAEGW-U selects an SFP.
3	SAEGW-U adds metadata as NSH context header and forwards it to the NSH supported appliance.
4	The NSH supported appliance sends back the packets to the SAEGW-U with the some metadata tags,as sent by SAEGW-U.
5	On receiving the packets, SAEGW-U classifies the subscriber data traffic that is based on subscriber charging policies.
6	SAEGW-U sends the data packets to the subscriber.

NSH Traffic Steering Requirements

Following is the behavior for integration of NSH appliances in the Traffic steering solution:

- SAEGW-U maintains the session stickiness of NSH appliance and ensure that all flows of a subscriber session end up selecting the same appliance instance.
- There’s a configurable option to define the load capacity for every appliance instance, example 50%, 100%. If the load status by the NSH appliance exceeds this threshold, only existing subscribers can continue with such instance. This instance doesn’t allocate to any new subscriber until the load status falls below the threshold.

- If NSH appliance detects as DEAD, all traffic on SFPs engaging this appliance instance is reclassified and traffic moves to a different appliance instance. Such appliance isn't available for new subscriber selection once it comes back ALIVE.
- Traffic Steering can be enable/disable in midsession. If you enable the traffic steering in between, then it's applicable to new flows. Old flows continue without traffic-steering.
- SR/ICSR support for traffic-steering Post SR/ICSR session stickiness is maintained.
- In case of multi appliance SFP, there are two forms of configurations:
  - For cases where appliances need to see start of traffic (example - TWH Packets), an SFP is selected which engages all appliances. As per the configuration policies, when the classification happens, the traffic can fall out of ineligible appliances.
  - For cases where appliances engage in mid flow, the configuration is such that appliances engage once the certain appliances become eligible further to traffic classification.
- Traffic steering statistics indicate the packets which are candidate for traffic steering. For traffic steering statistics, those packets are also counted which are dropped by quota exhaust, though they are the candidates for traffic steering.
- When node failure is done and continuous data is coming, then there can be discrepancy in steering statistics. Data steered on SFPs which is going down is not reflected in statistics.
- If you want to modify the NSH remote IP add or SRC bind IP in the configuration for any NSH appliance instance:
  - Then remove the instance.
  - Then remove the SFPs associated with it.
  - Put the SFPs and new instance with modified IP addresses.
  - Perform the commit afterwards.

This feature supports the following Traffic steering system limits:

Traffic steering object	Max Limit
Total Appliance groups	16
Totsl Instances per Appliance Group	256
Total SFCs	16
Total SFPs	64000

Default Service Chain

For operator, there could be certain use cases, where all traffic for a subscriber who has traffic steering enabled, needs to traverse through certain appliance(s). In order to cater to such requirement while providing an easy configuration mechanism to achieve that, the concept of a default service chain has been brought in. For e.g. if the subscriber is engaged on a subscriber with 2 appliances, APP1 and APP2 , where APP2 needs to see all the traffic, a service chain containing APP2 would be configured as the default service chain.

Thus, for a traffic steering enabled subscriber, there could be unavailability of service chain APP1+APP2 for certain traffic due the following conditions:

- There is no suitable policy configured for certain flows which would select the APP1+APP2 service chain.
- APP1+APP2 service chain was selected ,but APP1 instances went down below the min instance threshold. In such case the APP1+APP2 service chain will not be available.
- APP1+APP2 service chain was selected but no SFP could be selected.

Under such cases due to service chain unavailability, the flows would fall back to the configured default service chain thus ensuring APP2 service treatment to the flows.

If a default service chain, however, if not configured, will lead to the traffic being sent out non-steered.

SFP Selection

SFP selectios is based on the:

- MSISDN Stickiness (preconfigured) or
- Load Availability

MSISDN Stickiness

MSISDN Stickiness depends on the MS-ISDN and it provides the corresponding node. If the node is available and is part of the SFP, then that SFP is selected for the data (UL/DL). Presently, MSISDN stickiness is available for the L2 nodes only and there can be a service chain having L2 nodes alone or with a mix of L2 and NSH. All SFPs of the service chain have same set of type of nodes, where type can be of L2 or L2 + NSH or NSH (only).

Subscriber Stickiness (for both L2 and NSH) is maintained for the subscriber across the service chains till that node is available and when node goes down or removed from the config, subscriber can move to a different SFP (based on SFP selection). In case of stickiness miss, logs and traps are generated.

Load Availability

Load availability is load capacity, current load is maintained for each SFP (minimum of all instances that are part of the SFP). The SFPs are classified as part of available, overloaded or blocked list based on load availability. Only available-list and overloaded-list are being used for SFP selection as blocked-list is for SFPs for which node is down. Available-list SFPs are available for both old and new calls/sessions. Overloaded-list (load availability =0) is only used for maintaining the stickiness (if any), that is for old calls/sessions only. SFPs, once selected may move to overloaded-list because of load and for maintaining the stickiness. Same SFPs are used for the old calls/sessions and new calls use the remaining SFPs of the available-list for SFP selection.

Interworking with Inline Features

Support for interworking with the following inline features is not in the scope of the existing implementation.

- IPv4/v6 Readdressing
- NAT44 and NAT64

- Next Hop Forwarding
- L2 Marking

The encoding of rating group in the NSH context header is supported aligned with the following expected behaviour:

- The encoded rating group value corresponds to the rule that each packet matches. So, in a single flow's packets, the rating group either changes or is not encoded as the flow moves across different rules with different rating groups configured/or not configured.
- The SAEGW populates the rating group value, if configured, in the rating group field. If only content id is configured then this value is populated in the field. In the event that none are associated with the packet's matching rule, no TLV field corresponding to the rating group is sent.
- In case SAE-GW performs a deferred rule match and send out the packets without a rule match, it doesn't encode the rating group TLV for such packets.

Configuring the L2 and NSH Traffic Steering Feature—Standalone Mode

The following sections provide information about the CLI commands available to configure the L2 and NSH traffic steering CUPS feature in both CP and UP.

Configuring the Control Plane

Perform the following steps to configure the CP:

1. The following CLI command is a sample configuration to configure CP under the active-charging service.

```
configure
  active-charging service ACS
  policy-control services-framework

  trigger-action ta1
    up-service-chain sc_L3
  exit
  trigger-action ta2
    up-service-chain  L3
  exit

  trigger-condition tc1
    rule-name = rule1
    rule-name = rule2
    multi-line-or
  exit

  trigger-condition tc2
    any-match = TRUE
  exit

  service-scheme scheme1
    trigger rule-match-change
      priority 1 trigger-condition tc1 trigger-action ta1
    exit
    trigger subs-scheme-received
      priority 1 trigger-condition tc2 trigger-action ta2
```

```
    exit

    subs-class class1
      subs-scheme = s1
    exit
    subscriber-base base1
      priority 1 subs-class class1 bind service-scheme scheme1
    exit
end
```

NOTES:

- **subs-scheme**: The name should match the subscription-scheme AVP value that is received from PCRF over the Gx interface.
- **up-service-chain SecNet**: The value must match the up-service-chain that is configured on UP.
- **rule-name**: The value can be static/predef/gor/dynamic rules.

2. Traffic steering AVPs are currently supported with the Diameter dictionary custom44. The Diameter dictionary enables CP to properly decode the TS-related AVPs when they are received over the Gx interface and sent in Sx message to UP.

   The following is an example configuration to configure the Dictionary in CP.

```
configure
  context ISP1
    ims-auth-service IMSGx
    policy-control
    diameter dictionary dpca-custom44
  exit
end
```

Following are the sample values for TS-related AVPs received over GX in CCA-I/CCA-U/RAR.

```
     [V] Services:
       [V] Service-Feature:
         [V] Service-Feature-Type: TS (4)
         [V] Service-Feature-Status: ENABLE (1)
         [V] Service-Feature-Rule-Install:
           [V] Service-Feature-Rule-Definition:
             [V] Service-Feature-Rule-Status: ENABLE (1)
             [V] Subscription-Scheme: scheme
             [V] Profile-Name: Gold
```

Configuring the User Plane

Perform the following steps in same sequence to configure the UP:

The following CLI command is a sample configuration to add an interface in the contexts, which are used to send data toward L2 and NSH supported appliance.

1. Add the interface in the contexts which will be used to send data toward the L2 and NSH supported appliance.

   The following is a sample configuration:

```
configure
require tsmon
end
configure
   context ISP1-UP
   interface <ts_ingress>
```

```
   ip address <ip_address>
   ipv6 address <ipv6_address_secondary>
   exit
   end

configure
   context ISP2-UP
   interface <ts_egress>
   ip address <ip_address>
   ipv6 address <ipv6_address_secondary>
   exit
   end
```

2. Bind these newly-added interfaces to the physical ports of the UP.

   The following is an example configuration:

```
configure
   port ethernet 1/11
   vlan 1240
   no shutdown
   bind interface ts_ingress ISP1-UP
   exit
   exit
   port ethernet 1/12
   vlan 1240
   no shutdown
   bind interface ts_egress ISP2-UP
   exit
   exit
   end
```

3. Add the TS-related configuration in the UP.

   The following is an example configuration:

```
config

ts-bind-ip IP_UP01 ipv4-address 209.165.200.225 ipv6-address 4001::106

  nsh
node-monitor ipv4-address 209.165.200.226 ipv6-address 4001::107 poll-interval 1
retry-count 2 load-report-threshold 5 (node-monitor  is mandatory for NSH appliances,
default values are poll-interval=1, retry-count=2, load-report-threshold=5)
    up-nsh-format format1
      tag-value 250 imsi encode
      tag-value 66 msisdn encode
      tag-value 4 rating-group encode
      tag-value 1 stream-fp-md encode decode
      tag-value 2 reverse-stream-fp-md encode decode
      tag-value 76 subscriber-profile encode
      tag-value 3 secondary-srv-path-hdr encode
      tag-value 5 rat-type encode
      tag-value 51 mcc-mnc encode
      tag-value 255 apn encode
      tag-value 25 sgsn-address encode
    #exit
  #exit
  traffic-steering
    up-service-chain sc_L3
      sfp-id 9 direction uplink up-appliance-group L2 instance 1 up-appliance-group L3
instance 1
      sfp-id 10 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 1
      sfp-id 11 direction uplink up-appliance-group L2 instance 2 up-appliance-group L3
```

```
 instance 1
      sfp-id 12 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 2
      sfp-id 13 direction uplink up-appliance-group L2 instance 1 up-appliance-group L3
 instance 2
      sfp-id 14 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 1
      sfp-id 15 direction uplink up-appliance-group L2 instance 2 up-appliance-group L3
 instance 2
      sfp-id 16 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 2
      sfp-id 17 direction uplink up-appliance-group L2 instance 3 up-appliance-group L3
 instance 1
      sfp-id 18 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 3
      sfp-id 19 direction uplink up-appliance-group L2 instance 4 up-appliance-group L3
 instance 1
      sfp-id 20 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 4
      sfp-id 21 direction uplink up-appliance-group L2 instance 3 up-appliance-group L3
 instance 2
      sfp-id 22 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 3
      sfp-id 23 direction uplink up-appliance-group L2 instance 4 up-appliance-group L3
 instance 2
      sfp-id 24 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 4
    #exit
    up-service-chain L3
      sfp-id 1 direction uplink up-appliance-group L3 instance 1
      sfp-id 2 direction downlink up-appliance-group L3 instance 1
      sfp-id 3 direction uplink up-appliance-group L3 instance 2
      sfp-id 4 direction downlink up-appliance-group L3 instance 2
    #exit
    up-appliance-group L3
      steering-type nsh-aware
      up-nsh-format format4
      min-active-instance 1
      instance 1 ip address 40.40.40.3
      instance 2 ip address 40.40.40.4
    #exit
    up-appliance-group L2
      steering-type l2-mpls-aware
      min-active-instance 1
      instance 1 ingress slot/port 1/13 vlan-id 2136 egress slot/port 1/12 vlan-id 2136
  ingress-context ingress ip address 4101::1 egress-context egress ip address 4101::2
      instance 2 ingress slot/port 1/13 vlan-id 2137 egress slot/port 1/12 vlan-id 2137
  ingress-context ingress ip address 4201::1 egress-context egress ip address 4201::2
      instance 3 ingress slot/port 1/13 vlan-id 2138 egress slot/port 1/12 vlan-id 2138
  ingress-context ingress ip address 4301::1 egress-context egress ip address 4301::2
      instance 4 ingress slot/port 1/13 vlan-id 2139 egress slot/port 1/12 vlan-id 2139
  ingress-context ingress ip address 4401::1 egress-context egress ip address 4401::2
    #exit
```

4. Verify the above configurations using **show configuration** CLI command. Then, execute the **commit** CLI command for the configurations to be effective.

```
configure
  traffic-steering
    commit
end
```

Configuration Guidelines

This section describes the following guidelines that are required to properly configure the feature:

- Configure the TS-related configuration on UP in the same sequence as mentioned in the preceding sections. This method ensures that the interfaces used to steer traffic toward L2 are applied properly in the configuration.
- If the instance under up-appliance-group has to be modified or deleted, then all the associated sfp-id under up-service-chain must be removed or deleted first.
- If the preceding modification must be done to the associated instance and sfp-id after a call is initiated, then remove the sfp-ids and reconfigure them to avoid any issues.
- Apply any changes to the interface before configuring the up-appliance-group instance. If the changes to the interface are applied at a later stage, remove the up-service-chain configuration first and then the up-appliance-group configuration. After the interface modification is complete, reconfigure the service chain and appliance group.
- The entire UP service chain and appliance group must not be removed to remove an interface or sfpid.

N to M Traffic Steering

Following are the configuration steps for the N:M Traffic Steering:

1. Configure TS-bind ip in RCM host specific configuration for all active UPs.
2. Configure the required active charging ruledef, rulebase configurations and traffic steering configurations (up-nsh format, up-appliance-group and up-service-chain, commit CLI) in common configuration in RCM and do commit.
3. Reload the active and standby UP with Day-0 config which has require ts-mon, RCM config, node monitor CLI for L3 server monitoring, BFD related interfaces configuration for L2, and service schema config for traffic steering (trigger condition, trigger action and so on).
4. Check RCM pushes config to all UPs. Check all services are up on all the UPs.
5. Check that the VPP fastpath tables have SST, SSMT, and SST tables created. Also check global tables are created correctly.
6. Check the up-service-chain SFP status and make sure that the SFPs are in available state.

Configuration

Following are the sample configurations:

- **Day-0 config** : The following configurations are part of Day-0 config.
  - Require ts-mon and Node-monitor CLI to monitor L3 appliance as mentioned in the earlier configuration section. Each UP has its own physical IP to monitor L3 appliance.
  - BFD related interfaces configuration for L2. Vlan configuration and IP interface related configuration.
  - Service schema configuration (Trigger condition, service scheme and so on).

Note Optimisation is planned to move service schema configuration to common configuration. Currently if service scheme configuration needs to be modified then changes needs be done manually on all the UPs.

UP Sample Configuration

L3 Monitoring

```
config
require ts-mon
nsh
 node-monitor ipv4-address 209.165.200.227 poll-interval 5 retry-count 5
load-report-threshold 20
exit

    interface ISP1_TO_PDN
      ip address 209.165.200.227 255.255.255.224
      ipv6 address 4001::254/64 secondary
    #exit
```

Note on UP2, IP can be 40.40.40.454, this is physical IP address specific to that UP.

L2 Monitoring:

```
config
  context ingress
    bfd-protocol
      bfd multihop-peer 209.165.200.228 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.229 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.230 interval 50 min_rx 50 multiplier 20
    #exit
    interface TS_SecNet_v4 loopback
      ip address 209.165.200.231 255.255.255.224
    #exit
    interface TS_SecNet_v4_1 loopback
      ip address 209.165.200.232 255.255.255.224
    #exit
    interface TS_SecNet_v4_2 loopback
      ip address 209.165.200.233 255.255.255.224
    #exit
    interface TS_Secnet_ingress
      ip address 209.165.200.234 255.255.255.224
    #exit
    interface TS_Secnet_ingress1
      ip address 209.165.200.235 255.255.255.224
    #exit
    interface TS_Secnet_ingress2
      ip address 209.165.200.236 255.255.255.224
    #exit

    ip route static multihop bfd bfd1 209.165.200.231 209.165.200.228

    ip route static multihop bfd bfd2 209.165.200.232 209.165.200.229

    ip route static multihop bfd bfd3 209.165.200.233 209.165.200.230

    ip route 209.165.200.228 255.255.255.224 209.165.200.237 TS_Secnet_ingress
```

```
    ip route 209.165.200.229 255.255.255.224 209.165.200.238 TS_Secnet_ingress1

    ip route 209.165.200.230 255.255.255.224 209.165.200.239 TS_Secnet_ingress2

  #exit
end

config
  context egress
    bfd-protocol
    bfd multihop-peer 209.165.200.231 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.232 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.233 interval 50 min_rx 50 multiplier 20

    #exit
    interface TS_SecNet_v4 loopback
      ip address 209.165.200.228 255.255.255.224
    #exit
    interface TS_SecNet_v4_1 loopback
      ip address 209.165.200.229 255.255.255.224
    #exit
    interface TS_SecNet_v4_2 loopback
      ip address 209.165.200.230 255.255.255.224
    #exit
    interface TS_Secnet_egress
      ip address 209.165.200.237 255.255.255.224
    #exit
    interface TS_Secnet_egress1
      ip address 209.165.200.238 255.255.255.224
    #exit
    interface TS_Secnet_egress2
      ip address 209.165.200.239 255.255.255.224
    #exit
    subscriber default
    exit
    aaa group default
    #exit
    ip route static multihop bfd bfd4 209.165.200.228 209.165.200.231
    ip route static multihop bfd bfd5 209.165.200.229 209.165.200.232
    ip route static multihop bfd bfd6 209.165.200.230 209.165.200.233
    ip route 209.165.200.231 255.255.255.224 209.165.200.234 TS_Secnet_egress
    ip route 209.165.200.232 255.255.255.224 209.165.200.235 TS_Secnet_egress1
    ip route 209.165.200.233 255.255.255.224 209.165.200.236 TS_Secnet_egress2
  #exit
end
```

One sample interface configuraition to bind all interfaces to port and vlan.

```
port ethernet 1/11
    vlan 1608
      no shutdown
      bind interface TS_Secnet_ingress ingress
    #exit
    vlan 1609
      no shutdown
      bind interface TS_Secnet_ingress1 ingress
    #exit
    vlan 1610
      no shutdown
      bind interface TS_Secnet_ingress2 ingress
    #exit
  #exit
  port ethernet 1/13
    no shutdown
```

```
    vlan 1608
      no shutdown
      bind interface TS_Secnet_egress egress
    #exit
    vlan 1609
      no shutdown
      bind interface TS_Secnet_egress1 egress
    #exit
    vlan 1610
      no shutdown
      bind interface TS_Secnet_egress2 egress
    #exit
```

service schema configuration:

```
trigger-action ta1
      up-service-chain sc_L3
    #exit
    trigger-action default
      up-service-chain default
    #exit
    trigger-condition tc1
      rule-name = udp
      rule-name = http-pkts
      rule-name = tcp
      rule-name = dynamic2
      multi-line-or all-lines
    #exit
    trigger-condition tc2
      rule-name = qci8
      rule-name = qci1
      multi-line-or all-lines
    #exit
   trigger-condition defualt
     any-match = TRUE
   #exit
    service-scheme scheme1
      trigger rule-match-change
        priority 1 trigger-condition tc1 trigger-action ta1
        priority 2 trigger-condition tc2 trigger-action ta1
      #exit
      trigger subs-scheme-received
        priority 1 trigger-condition default trigger-action default
      #exit
     #exit
    subs-class class1
      subs-scheme = gold
    #exit
    subscriber-base base1
      priority 1 subs-class class1 bind service-scheme scheme1
    #exit
```

- **Host Specific configuration**: The following configurations is the part of the host specific configuration.
  - TS-bind IP configuration for each ACTIVE UP is the part of host specific configuration on RCM.

```
svc-type upinterface
redundancy-group 1
host Active1
host 391 " context ISP1-UP"
host 436 " interface ISP1_TO_PDN_v6 loopback"
host 437 " ipv6 address 4000::106/128"
host 438 " #exit"
host 439 " interface ISP1_TO_PDN_v4 loopback"
host 440 " ip address 209.165.200.240 255.255.255.224"
```

```
host 441 " #exit"
host 471 "ts-bind-ip up1 ipv4-address 209.165.200.240 ipv6-address 4000::106"
host 472 " exit"
host Active2
host 600 " context ISP1-UP"
host 601 " interface ISP1_TO_PDN_v6 loopback"
host 602 " ipv6 address 4000::107/128"
host 603 " #exit"
host 604 " interface ISP1_TO_PDN_v4 loopback"
host 605 " ip address 209.165.200.241 255.255.255.224"
host 606 " #exit"
host 607 "ts-bind-ip up2 ipv4-address 209.165.200.241 ipv6-address 4000::107"
host 608 " exit"
```

Note TS-bind IP is a loopback IP address. Its physical IP address is the part of Day-0 configuration.

- **Common configuration**: The following configuration is the part of the common configuration.
  - Traffic steering configuration (up-nsh format, up-appliance-group, and up-service-chain config).

```
nsh
    up-nsh-format L3-format
      tag-value 7 imsi encode
      tag-value 4 rating-group encode
      tag-value 1 stream-fp-md encode decode
      tag-value 2 reverse-stream-fp-md encode decode
      tag-value 76 subscriber-profile encode
      tag-value 3 secondary-srv-path-hdr encode
      tag-value 5 rat-type encode
      tag-value 51 mcc-mnc encode
      tag-value 255 apn encode
      tag-value 25 sgsn-address encode
    #exit

  #exit
  traffic-steering
  up-appliance-group L2
steering-type l2-mpls-aware
min-active-instance 1
instance 1 ingress slot/port 1/12 vlan-id 1608 egress slot/port 1/13 vlan-id 1608
ingress-context ingress ip address 209.165.200.231egress-context egress ip address
209.165.200.228 load-capacity 100
instance 2 ingress slot/port 1/12 vlan-id 1609 egress slot/port 1/13 vlan-id 1609
ingress-context ingress ip address 209.165.200.232egress-context egress ip address
209.165.200.229 load-capacity 80
instance 3 ingress slot/port 1/12 vlan-id 1610 egress slot/port 1/13 vlan-id 1610
ingress-context ingress ip address 209.165.200.233egress-context egress ip address
209.165.200.230 load-capacity 90
exit
    up-appliance-group L3_only
      steering-type nsh-aware
      up-nsh-format new
      min-active-instance 1
      instance 1 ip address 209.165.200.242 load-capacity 80
      instance 2 ip address 209.165.200.243 load-capacity 90
    #exit

    up-service-chain sc_L3
      sfp-id 1 direction uplink up-appliance-group L2 instance 1 up-appliance-group
```

```
L3_only instance 2
     sfp-id 2 direction downlink up-appliance-group L3_only instance 2 up-appliance-group
 L2 instance 1
      sfp-id 10 direction uplink up-appliance-group L2 instance 2 up-appliance-group
L3_only instance 2
      sfp-id 11 direction downlink up-appliance-group L3_only instance 2
up-appliance-group L2 instance 2
      sfp-id 12 direction uplink up-appliance-group L2 instance 3 up-appliance-group
L3_only instance 2
      sfp-id 13 direction downlink up-appliance-group L3_only instance 2
up-appliance-group L2 instance 3
      sfp-id 14 direction uplink up-appliance-group L2 instance 1 up-appliance-group
L3_only instance 1
      sfp-id 15 direction downlink up-appliance-group L3_only instance 1
up-appliance-group L2 instance 1
      sfp-id 16 direction uplink up-appliance-group L2 instance 2 up-appliance-group
L3_only instance 1
      sfp-id 17 direction downlink up-appliance-group L3_only instance 1
up-appliance-group L2 instance 2
      sfp-id 18 direction uplink up-appliance-group L2 instance 3 up-appliance-group
L3_only instance 1
      sfp-id 19 direction downlink up-appliance-group L3_only instance 1
up-appliance-group L2 instance 3
    #exit
 up-service-chain default
sfp-id 200 direction uplink up-appliance-group L3_only instance 1
sfp-id 201 direction downlink up-appliance-group L3_only instance 1
sfp-id 202 direction uplink up-appliance-group L3_only instance 2
sfp-id 203 direction downlink up-appliance-group L3_only instance 2
#exit
commit
exit
```

Show CLI for Verification

Following are the show CLIs for User Plane and RCM:

- User Plane: **Show srp checkpoints stats/ Show srp checkpoints stats debug-info**

  ```
  laas-setup# show srp checkpoint statistics | grep UPLANE_TRAFFIC_STEERING_INFO
  ```

- RCM : **under rcm checkpoint manager**

  ```
  "numTSInfo": 0
  ```

Monitoring and Troubleshooting—Standalone Mode

This section describes how to monitor and troubleshoot this feature.

Show Commands for Control Plane

This section describes the available show command to monitor this feature on CP.

show active-charging sessions full all

Note *TS Subscription Scheme Name*: Displays the subscription scheme that must be applied from the service-scheme configured under the active-charging-service. This active-charging-service is received from PCRF over the Gx interface.

Show Commands for User Plane

This section describes the available show commands to monitor this feature on UP.

Show Commands for Configuration

This section describes the available show commands to check configuration for the feature.

- **show user-plane-service traffic-steering up-service-chain all**
- **show user-plane-service traffic-steering up-service-chain name** *up-service-chain name*
- **show user-plane-service traffic-steering up-service-chain sfp-id** *sfp-id*

Show Commands for Data Statistics

This section describes the available show commands to check data statistics related to the feature.

- **show user-plane-service inline-services traffic-steering statistics up-service-chain all v**
- **show user-plane-service inline-services traffic-steering statistics up-service-chain all**
- **show user-plane-service inline-services traffic-steering statistics up-service-chain sfp-id** *sfp-id*
- **show user-plane-service inline-services traffic-steering statistics up-appliance-group all verbose**
- **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance group name*
- **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance group name* **instance** *appliance instance*

Show Commands to Check the Service Chain and SFP Association for TS:

This section describes the available show commands to check the service chain and SFP association.

- **show subscriber user-plane-only flows**
- **show subscribers user-plane-only callid** <*call-id*> **flows**

Show Commands for OAM Statistics

This section describes the available show commands to check OAM statistics related to the feature.

- **show user-plane-service inline-services traffic-steering oam all**
- **show user-plane-service inline-services traffic-steering oam summary**
- **show user-plane-service inline-services traffic-steering oam l3-steering summary**
- **show user-plane-service inline-services traffic-steering oam l3-steering monitors** <*ip address*>
- **show user-plane-service inline-services traffic-steering oam l3-steering monitors all**

- **show user-plane-service inline-services traffic-steering oam l3-steering monitors up-appliance-group** *<name>*
- **show user-plane-service inline-services traffic-steering oam l2-steering monitors all**
- **show user-plane-service inline-services traffic-steering oam l2-steering monitors up-appliance-group** *<name>*
- **show user-plane-service inline-services traffic-steering oam l2-steering summary**
- **clear user-plane-service traffic-steering oam statistics**
- **clear user-plane-service traffic-steering oam l3-steering statistics**

Currently BFD doesn't provide an API to clear session stats, so the following traffic-steering OAM clear command is extended to include l2-steering stats.

- clear user-plane-service traffic-steering
  - OAM - Clears OAM
  - statistics - Clears the User-Plane Traffic-steering Statistics
- clear user-plane-service traffic-steering OAM
  - L3-steering - Clear L3-steering OAM
  - statistics - Clears OAM statistics

Show Configuration Command

The following configuration is a snippet of a sample **show configuration** command for this feature.

```
  nsh
    up-nsh-format format4
      tag-value 250 imsi encode
      tag-value 66 msisdn encode
      tag-value 4 rating-group encode
      tag-value 1 stream-fp-md encode decode
      tag-value 2 reverse-stream-fp-md encode decode
      tag-value 76 subscriber-profile encode
      tag-value 3 secondary-srv-path-hdr encode
      tag-value 5 rat-type encode
      tag-value 51 mcc-mnc encode
      tag-value 255 apn encode
      tag-value 25 sgsn-address encode
    #exit
  traffic-steering
    up-service-chain L3
      sfp-id 65535 direction uplink up-appliance-group L3 instance 1
      sfp-id 65536 direction downlink up-appliance-group L3 instance 2
      sfp-id 65537 direction downlink up-appliance-group L3 instance 1
      sfp-id 65538 direction uplink up-appliance-group L3 instance 2
    #exit
    up-service-chain sc_L3
      sfp-id 9 direction uplink up-appliance-group L2 instance 1 up-appliance-group L3
instance 1
      sfp-id 10 direction downlink up-appliance-group L3 instance 1 up-appliance-group L2
instance 1
      sfp-id 11 direction uplink up-appliance-group L2 instance 2 up-appliance-group L3
```

```
instance 1
      sfp-id 12 direction downlink up-appliance-group L3 instance 1 up-appliance-group L2
instance 2
      sfp-id 13 direction uplink up-appliance-group L2 instance 1 up-appliance-group L3
instance 2
      sfp-id 14 direction downlink up-appliance-group L3 instance 2 up-appliance-group L2
instance 1
      sfp-id 15 direction uplink up-appliance-group L2 instance 2 up-appliance-group L3
instance 2
      sfp-id 16 direction downlink up-appliance-group L3 instance 2 up-appliance-group L2
instance 2
      sfp-id 17 direction uplink up-appliance-group L2 instance 3 up-appliance-group L3
instance 1
      sfp-id 18 direction downlink up-appliance-group L3 instance 1 up-appliance-group L2
instance 3
      sfp-id 19 direction uplink up-appliance-group L2 instance 4 up-appliance-group L3
instance 1
      sfp-id 20 direction downlink up-appliance-group L3 instance 1 up-appliance-group L2
instance 4
      sfp-id 21 direction uplink up-appliance-group L2 instance 3 up-appliance-group L3
instance 2
      sfp-id 22 direction downlink up-appliance-group L3 instance 2 up-appliance-group L2
instance 3
      sfp-id 23 direction uplink up-appliance-group L2 instance 4 up-appliance-group L3
instance 2
      sfp-id 24 direction downlink up-appliance-group L3 instance 2 up-appliance-group L2
instance 4
    #exit
    up-appliance-group L3
      steering-type nsh-aware
      up-nsh-format format4
      min-active-instance 1
      instance 1 ip address 209.165.200.225
      instance 2 ip address 4001::3
    #exit
    up-appliance-group L2
      steering-type l2-mpls-aware
      min-active-instance 1
      instance 1 ingress slot/port 1/13 vlan-id 2136 egress slot/port 1/12 vlan-id 2136
ingress-context ingress ip address 4101::1 egress-context egress ip address 4101::2
load-capacity 100
      instance 2 ingress slot/port 1/13 vlan-id 2137 egress slot/port 1/12 vlan-id 2137
ingress-context ingress ip address 4201::1 egress-context egress ip address 4201::2
load-capacity 60
      instance 3 ingress slot/port 1/13 vlan-id 2138 egress slot/port 1/12 vlan-id 2138
ingress-context ingress ip address 4301::1 egress-context egress ip address 4301::2
load-capacity 20
      instance 4 ingress slot/port 1/13 vlan-id 2139 egress slot/port 1/12 vlan-id 2139
ingress-context ingress ip address 4401::1 egress-context egress ip address 4401::2
load-capacity 100
    #exit
  #exit
  ts-bind-ip nshsrcip ipv4-address 209.165.200.226 ipv6-address 4001::106
  #exit
    context egress
    bfd-protocol
      bfd multihop-peer 4101::1 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 4201::1 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 4301::1 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 4401::1 interval 50 min_rx 50 multiplier 20
    #exit
    interface ts_egress1
      ipv6 address 4101::2/64
      ip mtu 1600
```

```
  #exit
  interface ts_egress2
    ipv6 address 4201::2/64
    ip mtu 1600
  #exit
  interface ts_egress3
    ipv6 address 4301::2/64
    ip mtu 1600
  #exit
  interface ts_egress4
    ipv6 address 4401::2/64
    ip mtu 1600
  #exit
  subscriber default
  exit
  aaa group default
  #exit
  gtpp group default
  #exit
  ipv6 route static multihop bfd bfd1 4101::2 4101::1
  ipv6 route static multihop bfd bfd2 4201::2 4201::1
  ipv6 route static multihop bfd bfd3 4301::2 4301::1
  ipv6 route static multihop bfd bfd4 4401::2 4401::1
  ip igmp profile default
  #exit
#exit
context ingress
  bfd-protocol
    bfd multihop-peer 4101::2 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4201::2 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4301::2 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4401::2 interval 50 min_rx 50 multiplier 20
  #exit
  interface ts_ingress1
    ipv6 address 4101::1/64
    ip mtu 1600
  #exit
  interface ts_ingress2
    ipv6 address 4201::1/64
    ip mtu 1600
  #exit
  interface ts_ingress3
    ipv6 address 4301::1/64
    ip mtu 1600
  #exit
  interface ts_ingress4
    ipv6 address 4401::1/64
    ip mtu 1600
  #exit
  subscriber default
  exit
  aaa group default
  #exit
  gtpp group default
  #exit
  ipv6 route static multihop bfd bfd1 4101::1 4101::2
  ipv6 route static multihop bfd bfd2 4201::1 4201::2
  ipv6 route static multihop bfd bfd3 4301::1 4301::2
  ipv6 route static multihop bfd bfd4 4401::1 4401::2
  ip igmp profile default
  #exit
#exit
  context ISP1-UP
  ip access-list IPV4ACL
```

```
    redirect css service ACS any
    permit any
  #exit
  ipv6 access-list IPV6ACL
    redirect css service ACS any
    permit any
  interface TO-ISP12
    ipv6 address 4001::106/64
    ip address 209.165.200.226 255.255.255.224 secondary
    ip mtu 2000
  #exit
    port ethernet 1/12
  no shutdown
  vlan 2135
    no shutdown
    bind interface TO-ISP12 ISP1-UP
  #exit
  vlan 2136
    bind interface ts_egress1 egress
  #exit
  vlan 2137
    no shutdown
    bind interface ts_egress2 egress
  #exit
  vlan 2138
    no shutdown
    bind interface ts_egress3 egress
  #exit
  vlan 2139
    no shutdown
    bind interface ts_egress4 egress
  #exit
#exit
port ethernet 1/13
  no shutdown
  vlan 2137
    no shutdown
    bind interface ts_ingress2 ingress
  #exit
  vlan 2138
    no shutdown
    bind interface ts_ingress3 ingress
  #exit
  vlan 2139
    no shutdown
    bind interface ts_ingress4 ingress
  #exit
  vlan 2136
    no shutdown
    bind interface ts_ingress1 ingress
  #exit
#exit
```

Show Command for User Plane 1:1 Redundancy

show srp checkpoint statistics | grep ts-sfp

```
call-recovery-uplane-internal-audit-ts-sfp-failure: 0
```

Show Commands for SFP availability

show user-plane traffic-steering up-service-chain <all> <name> <sfp-id>

SNMP Traps

The following SNMP Traps are added in support of this feature:

- UPlaneTsMisConfig : When there is no SFP that is associated with an appliance group.
- UPlaneTsNoSelectedSfp : When an SFP selection is not possible.
- UPlaneTsServiceChainOrApplianceDown : When a service chain or an application node becomes unavailable. The service chain is unavailable when the minimum instance of application group becomes unavailable.
- UPlaneTsServiceChainOrApplianceUp : When the node status of appliance is updated because the service chain or application node instance becomes available.

Bulk Statistics

Up-service-chain Schema

Variable Name	Data Type	Counter Type	Description
up-svc-chain-name	String	Info	Name of up service chain
up-svc-chain-status	Int32	Info	Status of up service chain
up-svc-chain-load-status	Int32	Gauge	Load status of up service chain
up-svc-chain-sfp-stickiness-miss-count	Int32	Counter	SFP stickiness miss count of up service chain
up-svc-chain-sfp-not-selected-count	Int32	Counter	SFP not selected count of up service chain
up-svc-chain-associated-calls	Int32	Gauge	Associated calls of up service chain
up-svc-chain-associated-flows	Int32	Gauge	Associated flows of up service chain
up-svc-chain-total-uplink-pkts	Int64	Counter	Total Uplink packets of up service chain
up-svc-chain-total-uplink-bytes	Int64	Counter	Total Uplink bytes of up service chain
up-svc-chain-total-downlink-pkts	Int64	Counter	Total Downlink packets of up service chain
up-svc-chain-total-downlink-bytes	Int64	Counter	Total Downlink bytes of up service chain

Up-appliance-group Schema

Variable Name	Data Type	Counter Type	Description
up-appl-group-name	String	Info	Name of up Appliance Group
up-appl-group-status	Int32	Info	Status of up appliance group
up-appl-group-load-status	Int32	Gauge	Load status of up appliance group
up-appl-group-node-down-count	Int32	Counter	Node down count of up appliance group
up-appl-group-associated-sfps	Int32	Gauge	Associated sfps of up appliance group
up-appl-group-num-times-loaded-state	Int32	Counter	Number of times node down state of up appliance group
up-appl-group-total-uplink-pkts	Int64	Counter	Total Uplink packets of up appliance group
up-appl-group-total-uplink-bytes	Int64	Counter	Total Uplink bytes of up appliance group
up-appl-group-total-downlink-pkts	Int64	Counter	Total Downlink packets of up appliance group
up-appl-group-total-downlink-bytes	Int64	Counter	Total Downlink bytes of up appliance group

The following CLI command is a sample bulkstats configuration for the feature.

```
config
   bulkstats collection
   bulkstats mode
   file 1
   up-service-chain schema TS format "\nup-service-chain-name = %up-svc-chain-name%
   \nup-service-chain-status=%up-svc-chain-status%\nup-service-chain-load-status =
   %up-svc-chain-load-status%\nup-service-chain-associated-calls =
   %up-svc-chain-associated-calls%\nup-service-chain-associated-flows =
   %up-svc-chain-associated-flows%\nup-service-chain-total-uplink-pkts =
   %up-svc-chain-total-uplink-pkts%\nup-service-chain-total-uplink-bytes =
   %up-svc-chain-total-uplink-bytes%\nup-service-chain-total-downlink-pkts =
   %up-svc-chain-total-downlink-pkts%\nup-service-chain-total-total-downlink-bytes
   = %up-svc-chain-total-downlink-bytes%\n\n"
```

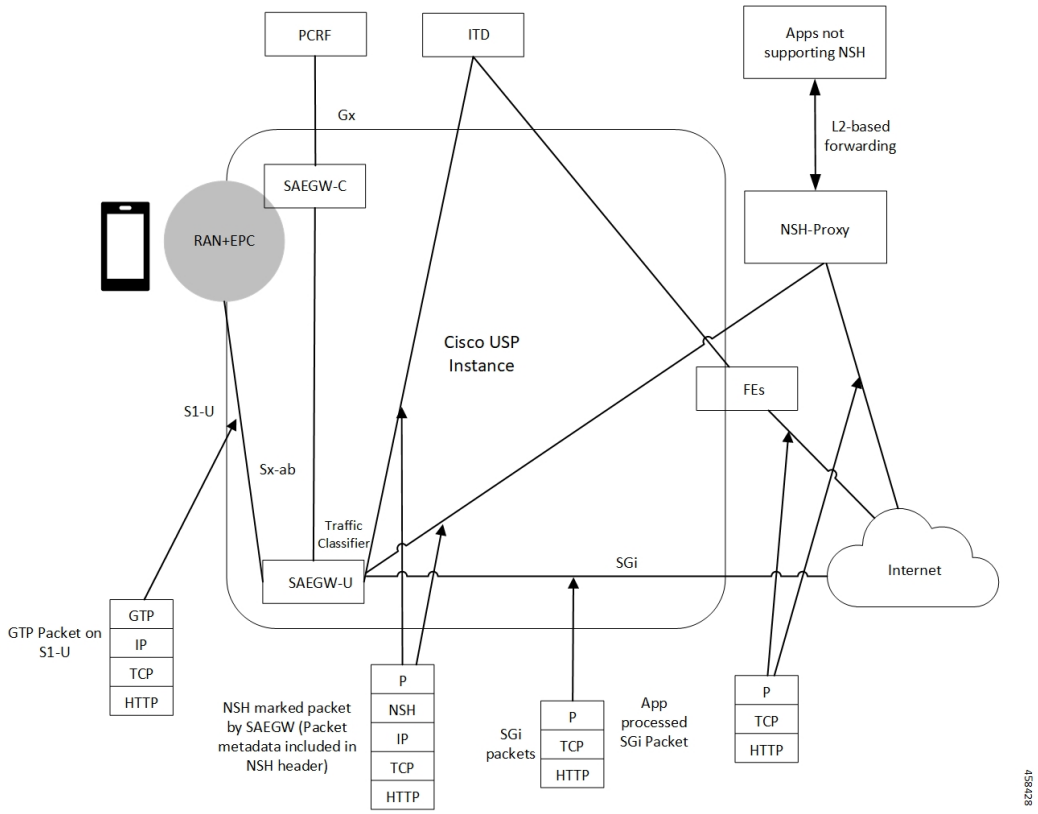
Feature Description—Sandwich Mode

The Sandwich Mode caters to the NSH-based Traffic Steering (TS) approach to provide the metadata needed by the service function appliance's Forwarding Engine (FE) nodes.

The Sandwich Mode solution leverages the Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director (ITD) in the Cisco USP instance. For more details about ITD, refer the *Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director Configuration Guide*.

Architecture—Sandwich Mode

The following figure illustrates the integration of an external service function appliance with Cisco’s SAEGW-U (User Plane).



The Sandwich Mode solution includes the following functionality:

- SAEGW-U adds the relevant NSH-Based-Metadata onto the relevant packets exiting the Gi path only in the Uplink direction.
- The ITD, running in Sandwich Mode may load-balance these packets (based on source-IP) to the FEs.
- SAEGW-U doesn't perform any health checks toward the FEs or aware of its existence.
- The ITD node may maintain the “stickiness” at a session level. The ITD does so by looking at the NSH-Outer-IP-SRC-Header.
- In the Uplink direction, the source IP is the "UE-IP" (Copy of Inner IP header). The destination IP is the "server-IP-internet".
- In the Downlink direction, there's no NSH Header and the packet straight away goes from the Internet into the FEs. At SAEGW-U, the source IP is the "Server-IP", and the destination IP is the "UE-IP".
- SAEGW-U performs the traffic classification and selects the service chain for a given flow.
- Service chain at SAEGW-U can include more than one appliance, and the steering functions can handle these appliances.
- SAEGW-U encodes only the NSH Header on Uplink packets.
- SAEGW-U copies the source IP details directly from the original UE-IP Header. SAEGW-U uses NSH Port 6633 for outer header SRC and DEST Port. The destination IP is the Appliance IP (as configured).
- On receiving Downlink packets with NSH header, the SAEGW-U drops such packets.

- SAEGW-U doesn't perform any health checks for the FEs or the ITD. The SAEGW-U treats the ITD as always available.
- SAEGW-U encodes all Uplink packets (qualified by the service function appliance) towards ITD with NSH Base Header, Service Headers, and Context Headers (with Metadata).
- TS App works only in one mode (either Sandwich mode or Standalone mode) at a time.

Note

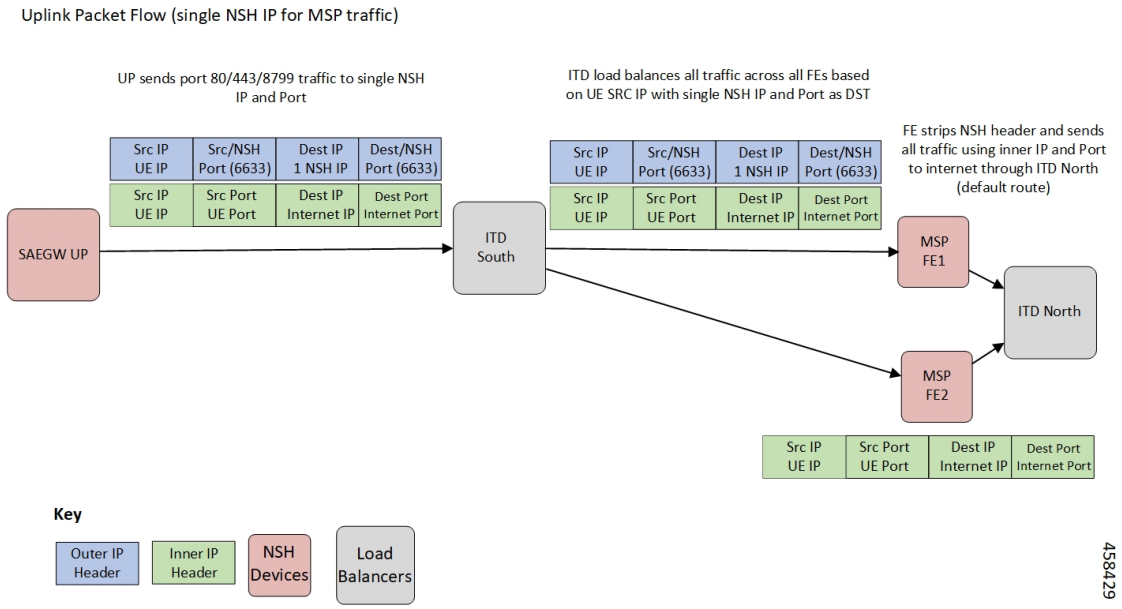
- For Sandwich mode, the **require tsmon** CLI command must not be configured.
- Changing from Sandwich to Standalone mode and conversely, requires a reboot.

How it Works—Sandwich Mode

Packet Flows in Sandwich Mode

Uplink Packets

The following figure illustrates the Uplink packet flow.



The following describes the packet flow:

1. GTP-U packet arrives at SAEGW-U. It decapsulates the GTP header and identifies the subscriber for the flow.

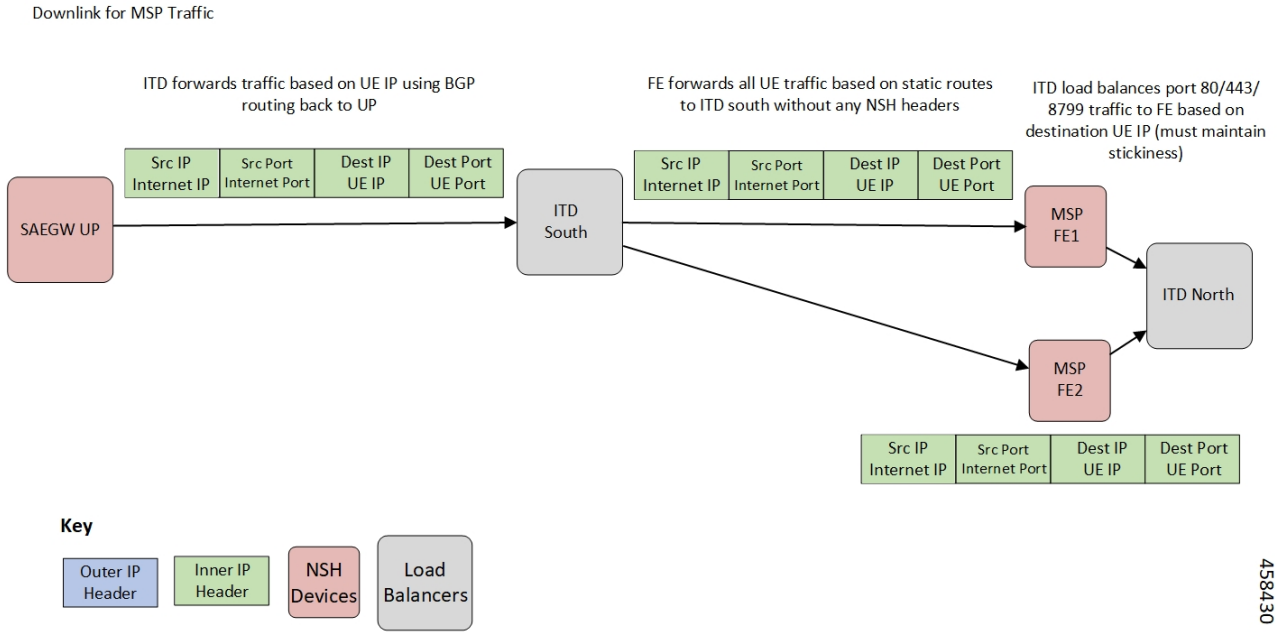
2. SAEGW-U performs traffic classification and associates a service chain for the flow. The SAEGW-U is configured to associate a service chain containing the service function appliance (ITD), with traffic classified depending on TCP/UDP/HTTP/HTTPS.
3. SAEGW-U looks up for NSH format associated with the service chain for encoding the parameters in the NSH variable header to be sent to the service function appliance.

   The following is an example of NSH Header with SFP selected for the Uplink packet is 200.

```
**************NSH Base Header**************
                    Version: 0
                    OAM Bit: 0
                     Length: 4
                    MD Type: 2
               Next Protocol: 1

**************NSH Service Header**************
   Service Path Identifier: 200
              Service Index:   1

**************Start NSH Context Header**************
                   TLV Type: <MSISDN tag configured in UP>
                   TLV Len: 15
                   TLV Value: 123456789012340 (unencrypted msisdn)

                   TLV Type: <MCCMNC tag configured in UP>
                   TLV Len: 6
                   TLV Value: 404122 (mcc-mnc value)

                   TLV Type: <RAT TYPE tag configured in UP>
                   TLV Len: 1
                   TLV Value: 3 (rat type value)

                   TLV Type: <APN tag configured in UP>
                   TLV Len: 64
                   TLV Value: APN1 (apn value)

                   TLV Type: <Sub Profile tag configured in UP>
                   TLV Len: 32
                   TLV Value: Profile-1 (Sub Profile name)

                   TLV Type: <SGSN addr tag configured in UP>
                   TLV Len: 4
                   TLV Value: 169090600 (SGSN Addr(in network byte order))


**************End NSH Context Header**************
```

Downlink Packets

The following figure illustrates the Downlink packet flow.



The following describes the packet flow:

1. Packets flow directly from internet server to the FEs. The FE processes the packets and sends it to the SAEGW-U.

   The SRC IP/Port is the server IP/Port and the DEST IP/Port is the UE IP/Port.

2. The SAEGW-U processes the packet, and if there are more service function appliances in the service chain, sends the packet for further processing. If the service chaining is complete, the packet is sent to normal Downlink packet processing path for Rule matching/classification and charging.

3. The SAEGW-U encapsulates the packet with GTP-U header and sends it across to the UE.

Note Downlink packets must not be NSH encoded. Otherwise, SAEGW-U will drop all such packets.

TCP and UDP Traffic

Uplink Traffic

- All TCP and UDP traffic qualified for steering towards the appliance is treated alike.
- UL packets are steered to the appliance with configured NSH context header elements. The NSH Service header is encoded with SI=1. Therefore, further to SI deduction and with SI=0, packet is sent over the Gi interface.
- The outer headers SRC IP is the same as the inner headers SRC IP (that is, UE SRC IP).
- The outer headers SRC Port is NSH port 6633.
- The outer headers DST IP is the configured Appliance IP.
- The outer headers DST PORT is the NSH port 6633.

Downlink Traffic

Downlink packets are received from FEs through ITD and therefore, processed as normal IP packet without being steered toward the FEs.

- UL packet received at the SAEGW-U is classified and based on the configured policy associated with the appropriate SFC.
- The SAEGW-U performs the SFP selection based on the service and load availability of the appliance instances and selected steer. The Uplink traffic is NSH (IP-UDP) encapsulated and steered on the selected SFP with the context header populated as deemed necessary.
- The NSH appliance on receiving the NSH packet, processes the IP packet (and possibly the context header), and sends the packet over the Gi interface.
- Downlink packet is sent by the destination server over the Gi interface to the SAEGW-U.

Service-Scheme Selection for Traffic Steering

You can select service-scheme in one of the following two ways:

1. Gx/PCRF:

   PCRF enables Traffic Steering through the following AVPs:

```
[V] Services:
  [V] Service-Feature:
    [V] Service-Feature-Type: TS (4)
    [V] Service-Feature-Status: ENABLE (1)
    [V] Service-Feature-Rule-Install:
      [V] Service-Feature-Rule-Definition:
        [V] Service-Feature-Rule-Status: ENABLE (1)
        [V] Subscription-Scheme: gold
        [V] Profile-Name: L3_profile
```

   TS Profile and TS Subscriber Scheme are then sent to User Plane through Sx messaging:

```
        SUBSCRIBER PARAMS:
...
...
...
            TS-Profile: L3_profile

            TS-Subscriber-Scheme: gold
```

   For Gx/PCRF based Traffic Steering, the **trigger subs-scheme-received** CLI command is required in the service-scheme configuration.

2. Service-scheme framework (without Gx/PCRF AVPs):

   Traffic Steering can be enabled without Subscription-scheme AVP from PCRF.

   The **trigger sess-setup** CLI command is required with trigger-action pointing to the **up-service-chain**. The following is an example configuration:

```
service-scheme scheme1
trigger sess-setup
  priority 1 trigger-condition subs-scheme-check trigger-action ta2
exit

trigger-condition subs-scheme-check
  any-match = TRUE
```

```
exit

trigger-action ta1
  up-service-chain SN-L3_profile

exit
```

Default Service Chain

For a TS-enabled subscriber, the following conditions can cause unavailability of service chain (APP1+APP2) for certain traffic:

- There's no suitable policy configured for certain flows which would select the APP1+APP2 service chain.
- APP1+APP2 service chain was selected, however, APP1 instances went down below the minimum instance threshold. In such case, the APP1+APP2 service chain won't be available.
- APP1+APP2 service chain was selected, however, no SFP could be selected.

Under such cases of service chain unavailability, the flows fall back to the configured default service chain and ensuring APP2 service treatment to the flows.

If a default service chain isn't configured, it leads to the traffic being sent out nonsteered.

For TS-enabled through Gx/PCRF, the default service chain is defined through **trigger subs-scheme-received**.

For TS-enabled through service scheme framework without Gx/PCRF AVPs, the default service chain is defined through **trigger sess-setup**.

SFP Selection

For service chains with only NSH-based appliance:

For Downlink packets, there's no NSH appliance and so, there's no SFP.

For service chains with a mix of L2 and NSH-based appliances:

Any SFP is selected based on L2 "stickiness". Same NSH-based appliance is present, and always available for SFP selection.

For Downlink packets, the SFP selection is based only on L2 appliance.

There's no SFP selection based on Load availability of NSH-based appliance. The NSH/appliance is considered as always-available.

Limitations and Restrictions

The following are the known limitation/restrictions of the feature:

- Changing from Standalone mode to Sandwich mode and vice versa, requires a reload and configuration change.
- When Traffic Steering is enabled from PCRF or locally using the service-scheme framework, then Traffic Steering can't be disabled on that session.
- For multi appliance service chain (L2 and L3 steering), the SFPs for V4 and V6 traffic are different. However, both SFPs maintains the L2 appliances MSISDN based stickiness.

Configuring NSH Traffic Steering—Sandwich Mode

This section provides information about the CLI commands available to configure NSH Traffic Steering—Sandwich Mode in both CP and UP

CP Configuration

Perform the following steps to configure the CP:

1. Configure the Active Charging Service configuration.

   The following is an example configuration:

```
configure
  active-charging service ACS
  policy-control services-framework

  trigger-action ta1
    up-service-chain sn-L3-sc  <<<< (This should match the up-service-chain configured
on UP)
  exit

  trigger-action ta2
    up-service-chain  L3-sc    <<<< (This should match the up-service-chain configured
on UP)
  exit

  trigger-condition tc1
    rule-name = rule1           <<<<< (This can be static/predef/gor/dynamic rules)
    rule-name = rule2
    multi-line-or
  exit

  trigger-condition tc2
    any-match = TRUE
  exit

  service-scheme scheme1
    trigger rule-match-change
      priority 1 trigger-condition tc1 trigger-action ta1
    exit
    trigger subs-scheme-received      <<<<< (For default service chain selection)
      priority 1 trigger-condition tc2 trigger-action ta2
    exit

    subs-class class1
      subs-scheme = gold  <<<<<<< (This name should match the subscription-scheme AVP
value received from PCRF over Gx)
    exit

    subscriber-base base1
      priority 1 subs-class class1 bind service-scheme scheme1
    exit
end
```

2. Traffic steering AVPs are currently supported with the Diameter dictionary custom44. The Diameter dictionary enables CP to properly decode the TS-related AVPs when they are received over the Gx interface and sent in Sx message to UP.

   The following is an example configuration to configure the Dictionary in CP.

```
configure
  context ISP1
    ims-auth-service IMSGx
    policy-control
    diameter dictionary dpca-custom44
  exit
end
```

The following are the sample values for TS-related AVPs received over Gx in CCA-I/CCA-U/RAR.

```
     [V] Services:
       [V] Service-Feature:
         [V] Service-Feature-Type: TS (4)
         [V] Service-Feature-Status: ENABLE (1)
         [V] Service-Feature-Rule-Install:
           [V] Service-Feature-Rule-Definition:
             [V] Service-Feature-Rule-Status: ENABLE (1)
             [V] Subscription-Scheme: gold
             [V] Profile-Name: L3
```

UP Configuration

Perform the following steps in same sequence to configure the UP:

1. Add the interface in the contexts which will be used to send data toward the Service chain appliances.

   The following is an example configuration:

```
configure
  context ISP1-UP
    interface ts_ingress
    ip address 209.165.200.225 255.255.255.224
    ipv6 address 4101::1/64 secondary
  exit
end

configure
  context ISP2-UP
    interface ts_egress
    ip address 209.165.200.225 255.255.255.224
    ipv6 address 4101::2/64 secondary
  exit
end
```

2. Bind these newly-added interfaces to the physical ports of the UP.

   The following is an example configuration:

```
configure
  port ethernet 1/11
    vlan 1240
      no shutdown
      bind interface ts_ingress ISP1-UP
    exit
  exit
  port ethernet 1/12
    vlan 1240
      no shutdown
      bind interface ts_egress ISP2-UP
    exit
  exit
end
```

3. Add the TS-related configuration in the UP.

   The following is an example configuration:

```
configure
  ts-bind-ip IP_UP01 ue-src-ip ipv4-address 209.165.200.225     <<<< See Notes below

  nsh
    up-nsh-format nfo
      tag-value 1  apn encode
      tag-value 2  imsi encode
      tag-value 3  mcc-mnc encode
      tag-value 4  msisdn encode
      tag-value 5  rat-type encode
      tag-value 10 rating-group encode
      tag-value 11 sgsn-address encode
      tag-value 12 subscriber-profile encode
    exit
  exit

  traffic-steering
    up-service-chain L3
      sfp-id 1 direction uplink up-appliance-group L3 instance 1
    exit

    up-service-chain sn_L3
      sfp-id 3  direction uplink   up-appliance-group L2 instance 1 up-appliance-group
L3 instance 1
      sfp-id 4  direction downlink up-appliance-group L2 instance 1
      sfp-id 5  direction uplink   up-appliance-group L2 instance 2 up-appliance-group
L3 instance 1
      sfp-id 6  direction downlink up-appliance-group L2 instance 2
      sfp-id 7  direction uplink   up-appliance-group L2 instance 3 up-appliance-group
L3 instance 3
      sfp-id 8  direction downlink up-appliance-group L2 instance 3
      sfp-id 9  direction uplink   up-appliance-group L2 instance 4 up-appliance-group
L3 instance 3
      sfp-id 10 direction downlink up-appliance-group L2 instance 4

    exit
    up-appliance-group L3
      steering-type nsh-aware
      up-nsh-format nfo
      min-active-instance 1
      instance 1 ip address 40.40.40.3
    exit
    up-appliance-group L2
      steering-type l2-mpls-aware
      min-active-instance 1
      instance 1 ingress slot/port 1/13 vlan-id 2136 egress slot/port 1/12 vlan-id 2136
  ingress-context ingress ip address 4101::1 egress-context egress ip address 4101::2
      instance 2 ingress slot/port 1/13 vlan-id 2137 egress slot/port 1/12 vlan-id 2137
  ingress-context ingress ip address 4201::1 egress-context egress ip address 4201::2
      instance 3 ingress slot/port 1/13 vlan-id 2138 egress slot/port 1/12 vlan-id 2138
  ingress-context ingress ip address 4301::1 egress-context egress ip address 4301::2
      instance 4 ingress slot/port 1/13 vlan-id 2139 egress slot/port 1/12 vlan-id 2139
  ingress-context ingress ip address 4401::1 egress-context egress ip address 4401::2
    exit
```

NOTES:

- **ts-bind-ip** *name* **ue-src-ip { ipv4-address** *ipv4_address* **| ipv6-address** *ipv6_address* **}**: Specifies the IP address of the UP interface from which packet is sent out toward ITD.

4. Verify the above configurations using **show configuration** CLI command. Then, execute the **commit** CLI command for the configurations to be effective.

```
configure
  traffic-steering
    commit
end
```

Configuring Post Processing Ruledef in Both Standalone and Sandwich Mode

up-service-chain trigger action is used with trigger condition in the configuration of post processing a ruledef in the rulebase for steering the traffic. A single post processing ruledef is defined with port numbers for HTTP, HTTPS and other protocols even when there are multiple charging ruledefs. This single post processing ruledef name is matched in the trigger condition which is used in traffic steering.

Use the following configuration to configure post processing of ruledef for steering traffic:

```
configure
   active-charging service service_name
      rulebase rulebase_name
        post-processing priority priority_number ruledef ruledef_name
charging-action charging_action_name
        end
```

Use the following configuration to configure the trigger condition in post processing ruledef:

```
configure
   trigger-condition trigger_condition_name
      rule-name rule_name
      post-processing-rule-name post_processing_rule_name
      end
```

Configuring BFD Instance Id Using Interface Name in UP Appliance Group

During traffic steering, in the **up-appliance-group**, the BFD instance id is configured using the interface name and IP configuration.

Use the following configuration to configure BFD instance id for steering traffic:

```
configure
   traffic-steering
      up-appliance-group up_appliance_group_name
         steering-type steering_type
        instance instance_id ingress slot/port slot_or_port_number vlan-id vlan_id
 egress slot/port slot_or_port_number vlan-id vlan_id ingress-context ingress
interface-name interface-name egress-context egress interface-name interface-name

        end
```

Note

- For any given L2 **up-appliance-group**, the BFD instance id is configured using the IP address or the **interface-name** for the particular **ingress** or **egress** using the corresponding interface names.
- Once the **up-appliance-group** configuration is complete for BFD monitoring using the **interface-name**, the BFD registration takes upto five minutes to complete.
- Once the BFD registration is successful, the IP address and the **interface-name** will be available in the **show user-plane traffic-steering up-appliance-group all** output.
- In case the IP address changes for any **interface-name** used in the **up-appliance-group** with BFD monitoring, then the **up-appliance-group** must be reconfigured.

Monitoring and Troubleshooting the NSH Traffic Steering—Sandwich Mode

This section provides information about the CLI commands available for monitoring and troubleshooting the feature.

For details about SNMP Traps, refer SNMP Traps, on page 601 section of this chapter.

For details about Bulk Statistics, refer Bulk Statistics, on page 601 section of this chapter.

Show Commands

This sections provides information about the show CLI commands that are available in support of the feature.

CP Commands

Use the following show CLI command in CP to monitor and troubleshoot the feature: **show active-charging sessions full all**

TS Subscription Scheme Name: Displays the subscription scheme that must be applied from the service-scheme configured under the active-charging-service. This active-charging-service is received from PCRF over the Gx interface.

UP Commands

Use the following show CLI commands in UP to monitor and troubleshoot the feature.

- Traffic Steering configuration check
  - **show user-plane-service traffic-steering up-service-chain all**
  - **show user-plane-service traffic-steering up-service-chain name** *up_service_chain_name*
  - **show user-plane-service traffic-steering up-service-chain sfp-id** *sfp_id*
  - **show user-plane traffic-steering up-appliance-group name** *name* **instance-id** *id*
  - **show user-plane traffic-steering up-appliance-group name** *name*

- **show user-plane traffic-steering up-appliance-group all**
- **show user-plane traffic-steering up-service-chain name** *name*
- **show user-plane traffic-steering up-service-chain sfp-id** *id*
- **show user-plane traffic-steering up-service-chain all**

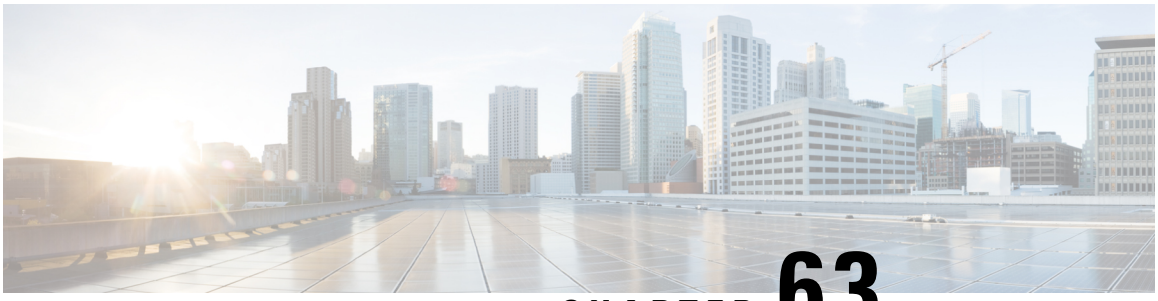
- Traffic Steering statistics
  - **show user-plane-service inline-services traffic-steering statistics up-service-chain all verbose**
  - **show user-plane-service inline-services traffic-steering statistics up-service-chain all**
  - **show user-plane-service inline-services traffic-steering statistics up-service-chain sfp-id** *sfp_id*
  - **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance_group_name*
  - **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance_group_ name* **instance** *appliance instance*
  - **show user-plane-service statistics trigger-action all**

- Service chain and SFP association
  - **show subscriber user-plane-only flows**
  - **show subscribers user-plane-only callid** *call_id* **flows**

show user-plane traffic-steering up-appliance-group all

Use the following show CLI command to monitor and troubleshoot the feature.

- **show in interface-name out interface-name**



CHAPTER 63

Packet Flow Description Management Procedure for Static and Predefined Rules

- Feature Description, on page 615
- How It Works, on page 615
- Monitoring and Troubleshooting, on page 626

Feature Description

The Packet Flow Description Management Procedure feature allows control plane to configure static and predefined rules and other charging information on the User Plane.

How It Works

Prior to CUPS, static and predefined rule processing was dependent on Rule-Def, Rule-Base, and Charging-Action. Rules-Base indicates the priority in which order static rules are needed to be matched and also provide associated charging action.

With the CUPS architecture, to process L3/L4 static and predefined rules, rule-def, rule-base, and charging-action need to be available at the User Plane. Using the PFD management message, control plane sends all this information to the associated User Plane.

To send this information from Control Plane to User Plane, CUPS architecture uses the following two modules:

- **Sx-U Demux:** Handles all node level messages with different Control-Plane nodes.
- **Sx-C Demux:** Handles node level message exchange with User-Plane service, that is, PFD management messages, Sx-association messages, Heartbeat related messages.

1. Once the Control-Plane is initialized with all the configuration and User-Plane is initialized with initial configuration, the PFD management request message is initiated using the debug mode CLI command. See to the *Monitoring and Troubleshooting* section for the debug command.
2. Once the debug CLI command is executed, the Sx-C demux pushes all the Rule-Def, Rule-Base, and Charging-Action configuration to the User-Plane using PFD management request or response message.

3. After the Sx-U demux on the User-Plane receives the PFD management request message, it decodes the configuration and sends it to each session manager instance at the User-Plane node and stores it in the SCT.

Moving Bulk Configurations from Control Plane to User Plane

A set of configurations can be pushed from the Control Plane (CP) to the User Plane (UP) using the **push config-to-up all** CLI command. A configuration timer constantly runs at the session controller. On expiration of this timer, various types of configurations in bulk are pushed to all designated session managers. The session controller maintains skip lists of various configuration types received from the CP. As and when the Sx Demux pushes the configuration, they are stored in skip lists based on the configuration type.

When the skip list reaches its maximum length, the entire list - for a particular configuration type, is pushed from the session controller to all session managers. This provision reduces the number of messenger events/messages between proclets as the configurations are sent in a single message rather than sending one message for each configuration.

The following configuration types supported for a bulk configuration push:

- Ruledef
- Charging Action
- Action Priority Lines
- Routing Rule configuration
- Group of Ruledef configuration
- Rule in Group of Ruledef configuration
- Rulebase L3/L4/L7 Info configuration
- APN configuration
- ACS Service configuration
- Service Chain configuration
- NSH Format
- NSH Field
- Traffic Steering Group
- Host pool configuration in ECS
- Port Map configuration in ECS
- Service scheme framework configuration in ECS
- X-header format in ECS
- Content filtering category Policy IDs in ECS

Currently, configuration propagation from CP to UP occurs only when Sx Association happens between CP and UP upon UP registration, or when **push config-to-up all** CLI is triggered. During configuration propagation, all the configurations are pushed from CP to UP. After UP registration occurs, when a new configuration is

added or existing configuration is modified, the UP has to be rebooted for registration to receive the updated configuration from CP. This is because configuration updates are not propagated to UP for now.

When the **push config-to-up all** CLI is executed, the entire configuration is propagated to all the registered/associated UPs. The configuration can be propagated to a specific UP as well by giving the peer address as input. The configuration is pushed only to UPs that are associated with the CP.

The **push config-to-up all** CLI does not delete any existing configuration on the UP and also does not flush out any unwanted configuration in UP, which is not present in CP. The configuration from CP merges with what is currently present in UP. The existing configuration on UP is not flushed out. The configuration audit between CP and UP is not supported.

Rule, Rulebase action priority, Host pool, and Port Map removal through configuration on CP leads to automatic push from CP to UP. Rule addition or modification requires push through the CLI.

Support for rule-lines modifications (addition or deletion) are added in the ruledef. The changed rule-lines are the candidate for rule matching for the existing flows, the new flows, or the new calls.

In CUPS (without RCM), modifications are done in Control Plane and pushed to User Plane via PFD mechanism. In CUPS (with RCM), changes are done in RCM and pushed to User Plane. Changes are done parallelly and separately in Control Plane.

The following table provides information about the impact of configuration change in new and existing calls.

Change in Configuration	Impact on existing calls (existing flows)	Impact on existing calls (new flows)	Impact on new calls
Existing Ruledef contents/New Rule addition	Rule match is enforced on existing flows after configuration change.	The configuration changes apply on new flows. For new flows, anyways fresh rule match would happen and the ruledef changes are applied on new flows for existing calls	The configuration changes apply on new calls. For new flows, anyways fresh rule match would happen and the ruledef changes are applied on flows for new calls.
No Ruledef	Rule in use cannot be deleted unless its action priority is deleted from the rulebase.	Rule match is enforced on existing flows after configuration change.	The configuration changes apply on new calls.
New Group of Ruledefs (GoR)/Changes to existing Group of Ruledefs contents (Add or Delete Rule in GoR)	Rule match is enforced on existing flows after configuration change.	The configuration changes apply on new flows. For new flows, anyways fresh rule match would happen and the GoR changes are applied on new flows for existing calls.	The configuration changes apply on new calls. For new flows, fresh rule match would happen, and the GoR changes are applied on flows for new calls.
No GoR	Rule in use cannot be deleted unless its action priority is deleted from the rulebase.	Rule match is enforced on existing flows after configuration change.	The configuration changes apply on new calls.

Change in Configuration	Impact on existing calls (existing flows)	Impact on existing calls (new flows)	Impact on new calls
No Rule in GoR	Rule match is enforced on existing flows after configuration change.	New flows go through a fresh rule match and configuration change takes effect.	New flows go through a fresh rule match and configuration change takes effect.
Action Priority Changes/Action Priority addition	Configuration changes apply on existing flows.	Configuration changes apply on new flows.	Configuration changes apply on new calls.
No Action Priority	Configuration changes apply on existing flows.	Configuration changes apply on new flows.	Configuration changes apply on new calls.
No-Rulebase	No-Rulebase is not supported.	No-Rulebase is not supported	No-Rulebase is not supported
No-APN	No-APN is not supported	No-APN is not supported	No-APN is not supported
IP source violation	No impact on existing calls	No impact on existing calls	Configuration changes apply on new calls.

Limitation

When CP is on VPC-DI, delay in PFD configuration push from CP to UP may be observed on systems with bulk configurations in CP that is connected to large number of UPs.

The delay is caused as VPC-DI is a multi-card chassis, with an inter-card communication process, that takes some time to fetch configurations from Shared/System Configuration Task (SCT) for each peer UP.

When CP is on VPC-SI, the delay is not observed.

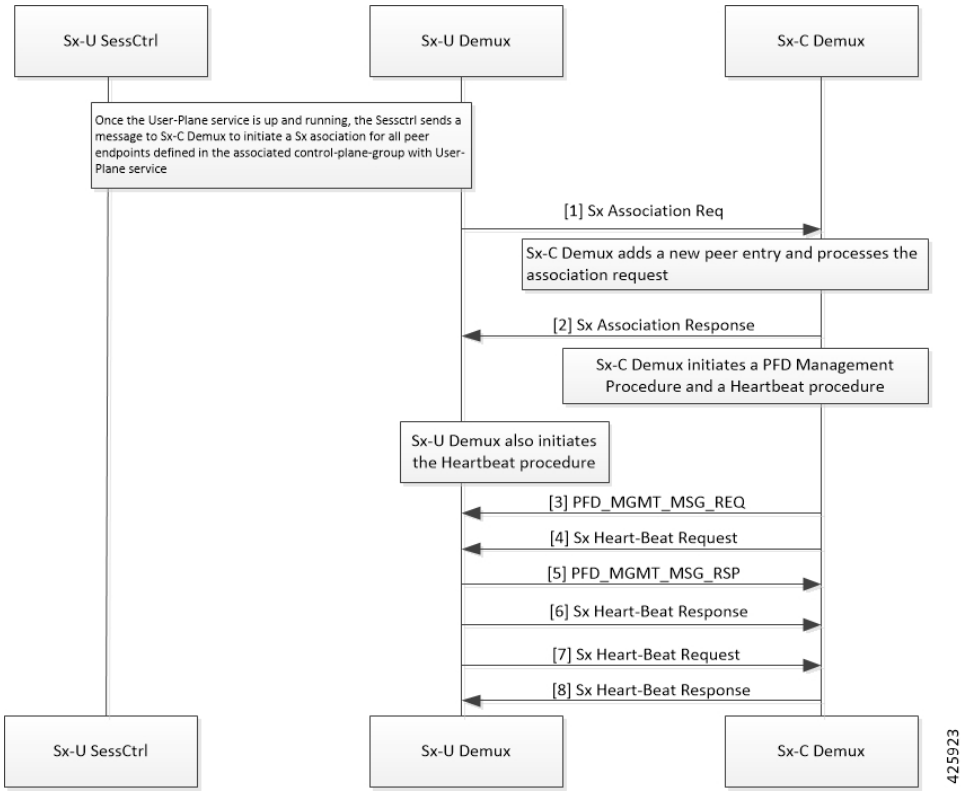
Sx Association

Important This feature is not fully qualified in this release. It is available only for testing purposes. For more information, contact your Cisco Account representative.

In CUPS environment, the Control Plane and User Plane entity should perform association with each other before establishing communication.

The Sx Association procedure is defined in 3GPP TS 29.244. As these are node level messages, they are handled by Sx-C Demux on Control Plane and Sx-U Demux on User Plane.

Call Flow



Following is a brief overview of how Sx Association works:

1. Sx Association Setup Request is initiated by Control Plane or User Plane.

Important In this release, only Sx Association Setup Request from User Plane is supported.

2. For User Plane to initiate Sx Association Setup Request, Operator should configure **control-plane-group** at Global Configuration mode and associate **control-plane-group** to User Plane Service. Refer *Configuring Sx Association Setup Request* section of this chapter.
3. Peer node ID (which is, currently, either IPv4 or IPv6 address) is configured in **control-plane-group**.
4. Currently, on User Plane, the Sx-U Demux uses Sx-Service Address (as it is on Node Id) which is sent into Sx Association Setup Request. Selection of IPv4 and IPv6 is depended on the configured **peer-node-id**.
5. After User Plane Service is up on User Plane, Sx-U Demux sends Sx Association Request toward Control Plane. Sx-C Demux validates and sends Sx Association Response toward User Plane.
6. After Control Plane processes Sx Association Request and sends response to User Plane, it starts Prime PFD message toward User Plane to send the configuration. Also, Control Plane starts Heartbeat procedure with the associated User Plane.
7. After receiving Sx Association Response, User Plane also starts Heartbeat procedure toward Control Plane.
8. If Control Plane is not ready (SAEGW Service is not up) when it receives Sx Association Setup Request, it rejects the Sx Association Setup Request. The User Plane reattempts Sx Association Set-up Request

after **association reattempt-timeout**. Refer the *Configuring Sx Association Reattempt Timeout* section of this chapter.

Release of Sessions for a Specific User Plane

To bring down a specific User Plane, it is recommended to first clear all subscribers that belong to that User Plane using the following CLI command:

clear subscribers saegw-only uplane-address *user_plane_address* **no-select-up**

Executing this CLI command releases all sessions that belong to the mentioned User Plane, gracefully, and marks that User Plane as "Not Available for Session Selection". This User Plane remains in Associated state but it will not be available for Session selection.

After clearing the session, execute either of the following CLI command on User Plane to remove its association from Control Plane.

no user-plane-service *service_name*

Or

no peer-node-id { ipv4-address *ipv4_address* **| ipv6-address** *ipv6_address* **}**

For additional information about the above CLI commands, refer *Configuring User Plane Service* and *Configuring Peer Node ID* sections in this guide.

To release only the existing sessions from a User Plane, use the following CLI command:

clear subscribers saegw-only uplane-address *user_plane_address*

In this case, note that the User Plane remains in Associated state and available for Session selection.

Note When the **clear subscribers** command is executed on UP, CP will not be informed and will consider the sessions as running.

ICSR Support

For Sx-Control Plane, Demux ICSR is supported. All associated Peer information is checkpointed to Standby Chassis Sx-Control Plane Demux through the Session manager.

Demux Recovery Support

Sx-Control Plane Demux recovery and unplanned Demux card migration is supported. During recovery, all associated Peer information is recovered from the Session manager to Sx-Control Plane Demux.

Currently, after Sx-Demux recovery, the Sx-Control Plane Demux does not perform audit with respective VPNmgr for peer entries and Peer ID. In case of any error, it can lead to call drop and out-of-sync situation between VPNMgr and SxMgr related to IP pool management and UP selection.

Configuring Control Plane Group

Use the following CLI commands to configure Control Plane Group under Global Configuration mode. The Control Plane Group lists the Control Plane endpoints to which the User Plane will be associated

```
configure
   [ no ] control-plane-group control_plane_group_name
   end
```

NOTES:

- **control-plane-group** *control_plane_group_name*: Configures Control Plane Group on User Plane. The *control_plane_group_name* should be a string of size 1 to 63.
- If previously configured, use the **no control-plane-group** *control_plane_group_name* CLI command to remove the Control Plane Group configuration

Configuring Sx Association

This sections describes the CLI commands available in support of this feature.

Configuring Sx Association Setup Request

Use the following CLI commands to enable the attributes related to Peer Node IDs and Sx Association under Control Plane Group Configuration mode.

```
configure
   control-plane-group control_plane_group_name
      sx-association { initiated-by-cp | initiated-by-up }
      end
```

NOTES:

- **sx-association**: Configures Sx Association Setup Request that is initiated by Control Plane or User Plane. The default value is **initiated-by-up**.
- **initiated-by-cp**: Sx Association Setup Request will be initiated by Control Plane.

Important This keyword is not supported in this release.

- **initiated-by-up**: Sx Association Setup Request will be initiated by User Plane.

Associating Control Plane Group with User Plane Service

Important Associating Control Plane Group with User Plane service is an optional parameter for User Plane service to come up. If there is Control Plane Group that is associated with User Plane, and as per its configuration it is supposed to start a Sx association, then the User Plane sends Sx Association Request to the defined Control Plane endpoint.

Use the following CLI commands to associate User Plane service with Control Plane Group.

```
configure
   contextcontext_name
      user-plane-service service_name
         [ no ] associate control-plane-group control_plane_group_name
         end
```

NOTES:

- **no**: Removes Control Plane Group association from User Plane service.
- **control-plane-group** *control_plane_group_name*: Associates Control Plane Group with which User Plane service performs Sx Association. The Control Plane Group name should be a string of size 1 to 63.

For more information about User Plane Service Configuration mode and it's relevant CLI commands, refer the *Configuring User Plane in CUPS* chapter.

Configuring Peer Node ID

Use the following CLI commands to configure Control Plane node IDs.

```
configure
   control-plane-group control_plane_group_name
     [ no ] peer-node-id { ipv4-address ipv4_address | ipv6-address ipv6_address
 }
     end
```

NOTES:

- **no**: Removes the followed option.
- **ipv4-address**: Configures IPv4 address.
- **ipv6-address**: Configures IPv6 address (supports colon-separated hexadecimal notation).
- The **peer-node-id** is the Control Plane sx-service address which should be started, and should receive and answer Setup requests.
- Currently, five node IDs can be added to the Control Plane group.

Configuring Sx Association Reattempt Timeout

Use the following configurations for Association Reattempt Timeout for Sx service.

```
configure
   context context_name
      sx-service service_name
         sx-protocol association reattempt-timeout timeout_seconds
         end
```

NOTES:

- **association**: Configures Sx Association parameters.
- **reattempt-timeout** *timeout_seconds*: Configures the Association Reattempt timeout for Sx Service, in seconds, ranging from 30 to 300. Default is 60.
- After User Plane starts, it waits for 2 minutes on SSI and 10 minutes on ASR 5500 to start Association Setup with Control Plane. This is done to make sure that the User Plane system is fully ready to handle configuration messages that are sent from the Control Plane after Association Setup. These values can be changed using reattempt-timeout.

Configuring Sx Association SNMP Traps

When an Sx association is detected, an SNMP trap (notification) is automatically generated by the system.

Use the following configuration to enable an SNMP trap when an Sx association is detected:

```
configure
   snmp trap enable SxPeerAssociated
   end
```

Use the following configuration to enable an SNMP trap when there is a Sx association release::

```
configure
   snmp trap enable SxPeerAssociationRelease
   end
```

Moving Bulk Configurations from Control Plane to User Plane

Use the following configuration to move bulk configurations from Control Plane to User Plane:

push config-to-up all peer-ip-addr*IP_Address*

NOTES:

- **all**: Pushes the configurations to all associated User Planes.
- **peer-ip-addr**: Pushes the configurations to a specified User Plane. The User Plane should be associated to receive the configuration.*IP_Address* (IPv4 or IPv6) specifies the IP address of the User Plane node.
- IP Pool related configurations are not pushed using the above configuration.

Monitoring and Troubleshooting Sx Association

This section provides information about CLI commands available for monitoring and troubleshooting the Sx Association procedure.

SNMP Trap

The following traps are available to track the status of an Sx Association:

- **sn_trap_sx_peer_node_associated**: An information trap which is triggered when an Sx association is detected. The following information is shared with both Control Plane and User Plane:
  - Context Name
  - Service Name
  - Node Type
  - Node ID
  - Peer Node Type
  - Peer Node ID
  - Group-Name
- **sn_trap_sx_peer_node_association_release**: An information trap which is triggered when an Sx association release is detected. The following information is shared with both Control Plane and User Plane:

- Context Name
- Service Name
- Node Type
- Node ID
- Peer Node Type
- Peer Node ID
- Group-Name

Show Commands and/or Outputs

This section provides information regarding show commands and/or their outputs in support of Sx Association.

show control-plane-group all

The ouput of this show command displays fields in support of Sx Association.

- Control Plane Group
  - Name:
  - Sx-Association:
  - Node-Id:
  - Node-Id:

show user-plane-service name <name>

The output of this show command displays the following fields in support of Sx Association.

- Service name
  - Service-Id
  - Context
  - Status
  - PGW Ingress GTPU Service
  - SGW Ingress GTPU Service
  - SGW Egress GTPU Service
  - Control Plane Tunnel GTPU Service
  - Sx Service
  - Control Plane Group

show sx peers

The output of this show command displays the fields in support of Sx Association.

- Node Type:
  - (C) - CPLANE
  - (U) - UPLANE
- Peer Mode:
  - (A) - Active
  - (S) - Standby
- Association State:
  - (i) - Idle
  - (I) - Initiated
  - (A) - Associated
  - (R) - Releasing
- Configuration State:
  - (C) - Configured
  - (N) - Not Configured
- IP Pool:
  - (E) - Enable
  - (D) - Disable
- Sx Service ID
- Group Name
- Node ID
- Peer ID
- Recovery Timestamp
- No of Restart
- Current Sessions
- Max Sessions

show snmp trap history

The output of this command includes the following fields:

- Timestamp
- Trap Information

Monitoring and Troubleshooting

This section provides information regarding the debug command and show commands and/or their outputs in support of this feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show user-plane-service charging-action all

This command displays the following output:

```
Service Name: default
Charging Action Name: charge-action-qci8
Content ID: 0
Service ID: 0
EDRs: Disabled
EGCDRs: Enabled
Rf: Disabled
UDRs: Enabled
Flow Idle Timeout: 300 (secs)
Limit For Flow Type: Disabled
Bandwidth ID: 0
Limit For Uplink Bandwidth: Disabled
Limit For Downlink Bandwidth: Disabled
Throttle-Suppress Timeout: n/a
QoS Renegotiate Traffic-Class: Disabled
QoS Class Identifier: 8
IP Type of Service: Not Configured
Content Filtering: Enabled
Credit-Control: Disabled
Flow Action:
Redirect URL: Disabled
Redirect URL from OCS: Disabled
Redirect to Video Server: Disabled
Clear Quota Retry Timer: Disabled
Conditional Redirect: Disabled
Discard: Disabled
Terminate-Flow: Disabled
Terminate-Session: Disabled
Rulebase Change: Disabled
Billing Action:
Event Data Record: Disabled
GGSN charging Data Record: Enabled
Rf Accounting: Disabled
User Data Record: Enabled
Radius Accounting Record: Disabled
Charge Volume: ip bytes
PCO-Custom1 value: n/a
Flow-Mapping Idle Timeout: 300 (secs)
DNS Proxy Bypass: Disabled
Discard on Readdressing Failure: Disabled
Video Bitrate: Not Configured (default/no(0) is interpreted as bitrate=QOS MBR (GGSN/PGW))
Strip URL:
CAE-Readdressing: Disabled
TFT notification to UE : Enabled
Service Detection:
Session Update:
```

```
QOS: Disabled
Packet Filter Name
==================
Predefined Rule Deactivation: Disabled
Config URRID : 0x800050
Charging Action Name: charge-action-qci9
Content ID: 0
Service ID: 0
EDRs: Disabled
EGCDRs: Enabled
Rf: Disabled
UDRs: Enabled
Flow Idle Timeout: 300 (secs)
Limit For Flow Type: Disabled
Bandwidth ID: 0
Limit For Uplink Bandwidth: Disabled
Limit For Downlink Bandwidth: Disabled
Throttle-Suppress Timeout: n/a
QoS Renegotiate Traffic-Class: Disabled
QoS Class Identifier: 8
IP Type of Service: Not Configured
Content Filtering: Enabled
Credit-Control: Disabled
Flow Action:
Redirect URL: Disabled
Redirect URL from OCS: Disabled
Redirect to Video Server: Disabled
Clear Quota Retry Timer: Disabled
Conditional Redirect: Disabled
Discard: Disabled
Terminate-Flow: Disabled
Terminate-Session: Disabled
Rulebase Change: Disabled
Billing Action:
Event Data Record: Disabled
GGSN charging Data Record: Enabled
Rf Accounting: Disabled
User Data Record: Enabled
Radius Accounting Record: Disabled
Charge Volume: ip bytes
PCO-Custom1 value: n/a
Flow-Mapping Idle Timeout: 300 (secs)
DNS Proxy Bypass: Disabled
Discard on Readdressing Failure: Disabled
Video Bitrate: Not Configured (default/no(0) is interpreted as bitrate=QOS MBR (GGSN/PGW))
Strip URL:
CAE-Readdressing: Disabled
TFT notification to UE : Enabled
Service Detection:
Session Update:
QOS: Disabled
Packet Filter Name
==================
Predefined Rule Deactivation: Disabled
Config URRID : 0x800050
Charging Action Name: ggsn-ingress
Content ID: 10
Service ID: 0
EDRs: Disabled
EGCDRs: Disabled
Rf: Disabled
UDRs: Enabled
Flow Idle Timeout: 300 (secs)
Limit For Flow Type: Disabled
```

```
Bandwidth ID: 0
Limit For Uplink Bandwidth: Disabled
Limit For Downlink Bandwidth: Disabled
Throttle-Suppress Timeout: n/a
QoS Renegotiate Traffic-Class: Disabled
QoS Class Identifier: Not Configured
IP Type of Service: Not Configured
Content Filtering: Enabled
Credit-Control: Disabled
Flow Action:
Redirect URL: Disabled
Redirect URL from OCS: Disabled
Redirect to Video Server: Disabled
Clear Quota Retry Timer: Disabled
Conditional Redirect: Disabled
Discard: Disabled
Terminate-Flow: Disabled
Terminate-Session: Disabled
Rulebase Change: Disabled
Billing Action:
Event Data Record: Disabled
GGSN charging Data Record: Disabled
Rf Accounting: Disabled
User Data Record: Enabled
Radius Accounting Record: Disabled
Charge Volume: ip bytes
PCO-Custom1 value: n/a
Flow-Mapping Idle Timeout: 300 (secs)
DNS Proxy Bypass: Disabled
Discard on Readdressing Failure: Disabled
Video Bitrate: Not Configured (default/no(0) is interpreted as bitrate=QOS MBR (GGSN/PGW))
Strip URL:
CAE-Readdressing: Disabled
TFT notification to UE : Enabled
Service Detection:
Session Update:
QOS: Disabled
Packet Filter Name
==================
Predefined Rule Deactivation: Disabled
Config URRID : 0x800050
Total charging action(s) found: 3
```

show user-plane-service charging-action name *charging-action-name*

This command displays the following output:

```
Charging Action Name: charge-action-qci1
Content ID: 0
Service ID: 0
EDRs: Disabled
EGCDRs: Enabled
Rf: Disabled
UDRs: Enabled
Flow Idle Timeout: 300 (secs)
Limit For Flow Type: Disabled
Bandwidth ID: 0
Limit For Uplink Bandwidth: Disabled
Limit For Downlink Bandwidth: Disabled
Throttle-Suppress Timeout: n/a
QoS Renegotiate Traffic-Class: Disabled
QoS Class Identifier: 1
IP Type of Service: Not Configured
```

```
Content Filtering: Enabled
Credit-Control: Disabled
Flow Action:
Redirect URL: Disabled
Redirect URL from OCS: Disabled
Redirect to Video Server: Disabled
Clear Quota Retry Timer: Disabled
Conditional Redirect: Disabled
Discard: Disabled
Terminate-Flow: Disabled
Terminate-Session: Disabled
Rulebase Change: Disabled
Billing Action:
Event Data Record: Disabled
GGSN charging Data Record: Enabled
Rf Accounting: Disabled
User Data Record: Enabled
Radius Accounting Record: Disabled
Charge Volume: ip bytes
PCO-Custom1 value: n/a
Flow-Mapping Idle Timeout: 300 (secs)
DNS Proxy Bypass: Disabled
Discard on Readdressing Failure: Disabled
Video Bitrate: Not Configured (default/no(0) is interpreted as bitrate=QOS MBR (GGSN/PGW))
Strip URL:
CAE-Readdressing: Disabled
TFT notification to UE : Enabled
Service Detection:
Session Update:
QOS: Disabled
Packet Filter Name
==================
Predefined Rule Deactivation: Disabled
Config URRID : 0x800050
Total charging action(s) found: 1
```

show user-plane-service rule-base all

This command displays the following output:

```
Service Name: default
Rule Base Name: prepaid
Charging Action Priorities:
Name Type Priority Charging-action Timedef Description
================================================================================
rule-qci8 RD 1 charge-action-qci8 - -
rule-qci7 RD 2 charge-action-qci7 - -
rule-qci6 RD 3 charge-action-qci6 - -
rule-qci5 RD 4 charge-action-qci5 - -
rule-qci4 RD 5 charge-action-qci4 - -
rule-qci3 RD 6 charge-action-qci3 - -
rule-qci2 RD 7 charge-action-qci2 - -
rule-qci1 RD 8 charge-action-qci1 - -
rule-qci9 RD 9 charge-action-qci9 - -
ip-any-rule RS 11 ggsn-ingress - -
Post-processing Action Priorities:
Name Type Priority Charging-action Description
================================================================================
Routing Action Priorities:
Ruledef Name Priority Analyzer Description
================================================================================
Groups of Prefixed Urls For Url Preprocessing :
EGCDR Fields:
Tariff time thresholds (min:hrs):
```

```
Interval Threshold : 0 (secs)
Uplink Octets : 0 Downlink Octets : 0
Total Octets : 0
Time Based Metering: Disabled
Content Filtering Group : Not configured
Content Filtering Policy : Not configured
Content Filtering Mode : Not configured
URL-Blacklisting Action : Not Configured
URL-Blacklisting Content ID : Not Configured
UDR Fields:
Interval Threshold : 0 (secs)
Uplink Octets : 0 Downlink Octets : 0
Total Octets : 0
First Hit Content-Id Trigger : Disabled
Tariff time trigger (min:hrs) : Disabled
NEMO-Prefix-Update Trigger : Disabled
CCA Fields:
RADIUS charging context: Not configured
RADIUS charging group : Not configured
RADIUS interim interval: Not configured
DIAMETER Requested Service Unit: Not configured
Quota Retry Time : 60 (secs)
Quota Holding Time (QHT): Not configured
Quota Time Duration Algorithms: Not configured
Flow End Condition : Disabled
Flow Any Error Charging Action: Disabled
Billing records : Disabled
Limit For Total Flows : Disabled
Limit For TCP Flows : Disabled
Limit For Non-TCP Flows : Disabled
FW-and-NAT Default Policy : n/a
PCP Service : n/a
QoS Renegotiation Timeout : Disabled
EDRs on DCCA Failure Handling : Disabled
EDRs on transaction complete : Disabled
Extract host from uri: Disabled
Tethering Detection : Disabled
OS-based Detection : N/A
UA-based Detection : N/A
Tethering Detection (ip-ttl) : Disabled
Max SYN detection in a flow : N/A
Tethering Detection (DNS-Based): Disabled
Tethering Detection (Application): Disabled
Websocket Flow-Detection Configuration:
n/a
Check-point Account Synchronization Timer Configuration:
SR : n/a
ICSR : n/a
EDR Suppress zero byte records : Disabled
EDR Timestamp Rounding : Round Off
EDR Charge Volume (sn-charge-volume)
Retransmissions counted : Enabled
Dropped counted : Disabled
EGCDR Timestamp Rounding : Round Off
RTP Dynamic Routing : Disabled
Ignore port number in application headers: Disabled
RTSP Delayed Charging : Disabled
Delayed Charging : Disabled
No Rating Group Override
No Service Id Override
IP Reassembly-Timeout : 5000 milliseconds
IP Reset ToS field : Disabled
IP Readdress Failure Terminate : Disabled
TCP Out-of-Order-Timeout : 5000 milliseconds
```

```
TCP Out-of-Order-Max-Entries : 1000 packets
TCP 2MSL Timeout : 2 sec Port Reuse: No
HTTP header parse limit : Disabled
RTSP initial bytes limit : Disabled
Xheader Certificate Name :
Xheader Re-encryption Period : 0 min
TCP MSS Modification : Disabled
TCP Check Window Size : Disabled
WTP Out-of-Order-Timeout : 5000 milliseconds
TCP transmit-out-of-order-packets : Immediately
WTP transmit-out-of-order-packets : Immediately
Verify Transport layer checksum : Enabled
ICMP Request Threshold : 20
Default Bandwidth-Policy : n/a
Bandwidth-Policy Fallback : Disabled
P2P Dynamic Routing : Disabled
TCP Proxy Mode Configuration:
TCP Proxy Mode : Disabled
CAE-Readdressing : Disabled
Transactional-Rule-Matching : Disabled
TRM Fastpath : Disabled
Override Control : Disabled
Override-Control-with-name : Disabled
Override-Control-with-grp-info : Disabled
Charging-Action Override : Disabled.
TFT notification to UE for default bearer : Enabled
Ran-Bandwidth Optimization : Disabled
Total rulebase(s) found: 1
```

show user-plane-service rule-base name *rule-base-name*

This command displays the following output:

```
Service Name: default
Rule Base Name: prepaid
Charging Action Priorities:
Name Type Priority Charging-action Timedef Description
================================================================================
rule-qci8 RD 1 charge-action-qci8 - -
rule-qci7 RD 2 charge-action-qci7 - -
rule-qci6 RD 3 charge-action-qci6 - -
rule-qci5 RD 4 charge-action-qci5 - -
rule-qci4 RD 5 charge-action-qci4 - -
rule-qci3 RD 6 charge-action-qci3 - -
rule-qci2 RD 7 charge-action-qci2 - -
rule-qci1 RD 8 charge-action-qci1 - -
rule-qci9 RD 9 charge-action-qci9 - -
ip-any-rule RS 11 ggsn-ingress - -
Post-processing Action Priorities:
Name Type Priority Charging-action Description
================================================================================
Routing Action Priorities:
Ruledef Name Priority Analyzer Description
================================================================================
Groups of Prefixed Urls For Url Preprocessing :
EGCDR Fields:
Tariff time thresholds (min:hrs):
Interval Threshold : 0 (secs)
Uplink Octets : 0 Downlink Octets : 0
Total Octets : 0
Time Based Metering: Disabled
Content Filtering Group : Not configured
Content Filtering Policy : Not configured
Content Filtering Mode : Not configured
```

```
URL-Blacklisting Action : Not Configured
URL-Blacklisting Content ID : Not Configured
UDR Fields:
Interval Threshold : 0 (secs)
Uplink Octets : 0 Downlink Octets : 0
Total Octets : 0
First Hit Content-Id Trigger : Disabled
Tariff time trigger (min:hrs) : Disabled
NEMO-Prefix-Update Trigger : Disabled
CCA Fields:
RADIUS charging context: Not configured
RADIUS charging group : Not configured
RADIUS interim interval: Not configured
DIAMETER Requested Service Unit: Not configured
Quota Retry Time : 60 (secs)
Quota Holding Time (QHT): Not configured
Quota Time Duration Algorithms: Not configured
Flow End Condition : Disabled
Flow Any Error Charging Action: Disabled
Billing records : Disabled
Limit For Total Flows : Disabled
Limit For TCP Flows : Disabled
Limit For Non-TCP Flows : Disabled
FW-and-NAT Default Policy : n/a
PCP Service : n/a
QoS Renegotiation Timeout : Disabled
EDRs on DCCA Failure Handling : Disabled
EDRs on transaction complete : Disabled
Extract host from uri: Disabled
Tethering Detection : Disabled
OS-based Detection : N/A
UA-based Detection : N/A
Tethering Detection (ip-ttl) : Disabled
Max SYN detection in a flow : N/A
Tethering Detection (DNS-Based): Disabled
Tethering Detection (Application): Disabled
Websocket Flow-Detection Configuration:
n/a
Check-point Account Synchronization Timer Configuration:
SR : n/a
ICSR : n/a
EDR Suppress zero byte records : Disabled
EDR Timestamp Rounding : Round Off
EDR Charge Volume (sn-charge-volume)
Retransmissions counted : Enabled
Dropped counted : Disabled
EGCDR Timestamp Rounding : Round Off
RTP Dynamic Routing : Disabled
Ignore port number in application headers: Disabled
RTSP Delayed Charging : Disabled
Delayed Charging : Disabled
No Rating Group Override
No Service Id Override
IP Reassembly-Timeout : 5000 milliseconds
IP Reset ToS field : Disabled
IP Readdress Failure Terminate : Disabled
TCP Out-of-Order-Timeout : 5000 milliseconds
TCP Out-of-Order-Max-Entries : 1000 packets
TCP 2MSL Timeout : 2 sec Port Reuse: No
HTTP header parse limit : Disabled
RTSP initial bytes limit : Disabled
Xheader Certificate Name :
Xheader Re-encryption Period : 0 min
TCP MSS Modification : Disabled
```

```
TCP Check Window Size : Disabled
WTP Out-of-Order-Timeout : 5000 milliseconds
TCP transmit-out-of-order-packets : Immediately
WTP transmit-out-of-order-packets : Immediately
Verify Transport layer checksum : Enabled
ICMP Request Threshold : 20
Default Bandwidth-Policy : n/a
Bandwidth-Policy Fallback : Disabled
P2P Dynamic Routing : Disabled
TCP Proxy Mode Configuration:
TCP Proxy Mode : Disabled
CAE-Readdressing : Disabled
Transactional-Rule-Matching : Disabled
TRM Fastpath : Disabled
Override Control : Disabled
Override-Control-with-name : Disabled
Override-Control-with-grp-info : Disabled
Charging-Action Override : Disabled.
TFT notification to UE for default bearer : Enabled
Ran-Bandwidth Optimization : Disabled
Total rulebase(s) found: 1
```

show user-plane-service rule-def all

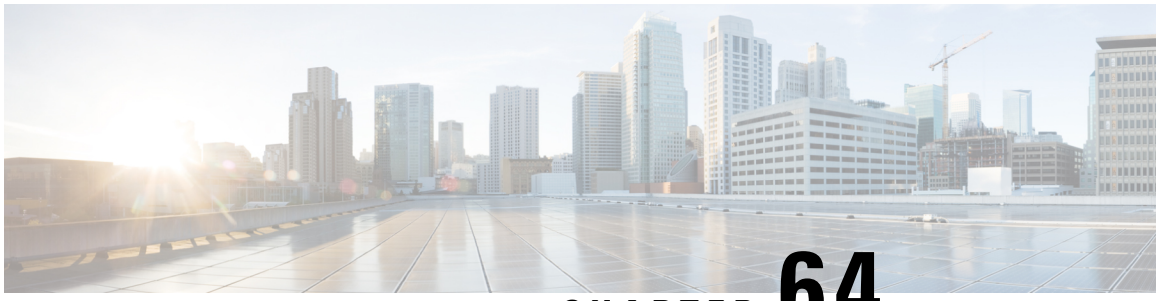
This command displays the following output:

```
Service Name: default
Ruledef Name: ip-any-rule
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Ruledef Name: rule-qci1
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Ruledef Name: rule-qci2
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Ruledef Name: rule-qci3
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Ruledef Name: rule-qci4
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Ruledef Name: rule-qci5
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Ruledef Name: rule-qci6
```

```
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Ruledef Name: rule-qci7
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
```

show user-plane-service rule-def name *rule-def-name*

```
Service Name: default
Ruledef Name: rule-qci8
ip any-match = TRUE
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: Disabled
Total Ruledef(s) : 1
```



CHAPTER 64

PDI Optimization

- Feature Summary and Revision History, on page 635
- Feature Description, on page 635
- How It Works, on page 636
- Configuring the PDI Optimization Feature, on page 641
- PDI Optimization OAM Support, on page 642

Feature Summary and Revision History

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The Packet Detection Information (PDI) Optimization feature allows the optimization of PFCP signaling, through Sx Establishment and Sx Modification messages, between the Control Plane and the User Plane function. Without PDI Optimization, the following common parameters are repeated in the PDI of all Packet Detection Rules (PDRs), for a given bearer, resulting in an unwanted increase in signaling between Control Plane and User Plane:

- Local F-TEID
- Network Instance
- UE IP address

- The PDI Optimization is achieved by consolidating the common parameters, in the PDI of the PDRs, into a single container that is called the Traffic Endpoint (Traffic Endpoint ID). The consolidated parameters from multiple PDRs are then referred to the Traffic Endpoint.
- The PDI Optimization is a CLI-controlled feature, and supported over the Sxa, Sxb, Sxc, Sxab, and N4 interfaces.

Relationships

The PDI Optimization feature is a prerequisite for the following features:

- GTP-U Error Indication Support on User Plane.
- Sx Bulkstats
- CUPS Bulkstats Support

How It Works

The Traffic Endpoint ID is unique within a PFCP session. When a PDI refers to a Traffic Endpoint, the parameters that are in the Traffic Endpoint is not provided in the PDI once again. The Control Plane function updates the Traffic Endpoint whenever applicable.

If a Traffic Endpoint is updated, all the PDRs that refer to this Traffic Endpoint in the User Plane function uses the updated information.

If the F-TEID allocation is performed in the User Plane function, the User Plane function allocates and stores the F-TEID associated to the Traffic Endpoint. When the User Plane function provides the allocated F-TEID to the Control Plane function in the PFCP Session Establishment response or PFCP Session Modification response message, the Control Plane function updates the Traffic Endpoint information that is stored in the Control Plane function with the received F-TEID.

The Control Plane function uses the Traffic Endpoint ID created in a different PFCP message only after getting the confirmation from the User Plane function of the Traffic Endpoint ID creation.

If the Control Plane function deletes a Traffic Endpoint, the User Plane function deletes all the PDRs that refer to the Traffic Endpoint that was deleted by Control Plane function. For Evolved Packet Core (EPC), the Remove Traffic Endpoint IE is used to delete a bearer for which multiple PDRs exist (with the same Traffic Endpoint ID).

The Traffic Endpoints is used as a mechanism to identify the bearers uniquely for a given Sx session on the User Plane. This is achieved with the help of Traffic Endpoint IDs that are associated with the PDRs of a bearer.

PDI Optimization Changes on Control Plane

A new container, called Traffic Endpoint, is supported to carry the repeated PDI information of a given bearer. Each Traffic Endpoint is associated with a Traffic Endpoint ID. This ID is unique for a given Sx Session.

A new IE, Create Traffic Endpoint IE, is supported as part of Sx Establishment Request.

Following are the new IEs supported as part of Sx Modification Request:

- Create Traffic Endpoint IE
- Update Traffic Endpoint IE
- Remove Traffic Endpoint IE

Create PDR supports a new IE, Traffic Endpoint ID, that identifies either the ingress or the egress Traffic Endpoint of a bearer to which this PDR is associated.

A new IE, Created Traffic Endpoint IE, is supported as part of Sx Establishment Response and Sx Modification Response message.

Create Traffic Endpoint IE

Following are the IEs in a Create Traffic Endpoint IE that are supported for a Pure-P call:

- Traffic Endpoint ID
- Local F-TEID
- Network instance
- UE IP address

Following are the IEs in a Create Traffic Endpoint IE that are supported for a Pure-S call:

- Traffic Endpoint ID
- Local F-TEID

NOTE: The Network instance and UE IP address IEs are currently not supported for a Pure-S call.

For a Collapsed call, Sxa Traffic Endpoints has IEs that are relevant to S-GW and Sxb Traffic Endpoints has IEs that are relevant to P-GW.

In addition to the 3GPP standards defined IEs, a private IE called "Bearer Info IE", is added to the Create Traffic Endpoint which includes:

- QCI of the bearer being created.
- ARP of the bearer being created.
- Charging ID of the bearer being created.

For a Pure-S call, there are two Traffic Endpoints that are created for each bearer of that PDN:

1. Create Traffic Endpoint for Ingress Traffic Endpoint, that is sent for the ingress F-TEID and referred by ingress S-GW PDR of the bearer.
2. Create Traffic Endpoint for Egress Traffic Endpoint, that is sent for the egress F-TEID and referred by egress S-GW PDR of the bearer.

For a Pure-S call, a bearer is uniquely identified on the User Plane that is based on Ingress and Egress Traffic Endpoint IDs of the bearer. The Traffic Endpoints also store the QCI, ARP, and Charging ID of the bearer.

For a Pure-P call, only one Traffic Endpoint is created for each bearer of that PDN. Create Traffic Endpoint for Ingress Traffic Endpoint, that is sent for ingress F-TEID and referred by ingress PDRs of the bearer. There is no separate egress Traffic Endpoint that is created for a Pure-P call as no Tunnel Endpoint ID is allocated on the P-GW egress. The same Traffic Endpoint is referred by both ingress and egress PDRs of a bearer. A

bearer is uniquely identified on the User Plane that is based on the Traffic Endpoint ID of the bearer. The Traffic Endpoint also stores the QCI, ARP, and Charging ID of the bearer.

For a Collapsed call, there are two Traffic Endpoints that are created for the S-GW leg of the call for each bearer. So, two Create Traffic Endpoints are sent for Ingress and Egress. The Sxa PDRs refer to these traffic endpoints based on the direction (ingress or egress). Only one Traffic Endpoint is created for the P-GW leg of the call for each bearer. The same Traffic Endpoint ID is referred by all Sxb PDRs of the bearer. For P-GW, Create Traffic Endpoint is sent for the ingress. The Traffic Endpoint IDs of Sxa and Sxb PDRs identify the bearer.

Created Traffic Endpoint IE

This IE is present in Sx Establishment/Sx Modification Response to inform Control Plane about the F-TEIDs that were locally allocated by the User Planes for the various Traffic Endpoints that were created.

Following are the IEs in a Created Traffic Endpoint IE:

- Traffic Endpoint ID
- Local FTEID

The information that is received in Created Traffic Endpoint IE is processed by the Control Plane, and the F-TEIDs that are allocated by the User Plane are stored in the Control Plane for ingress and egress accordingly.

Update Traffic Endpoint IE

This IE is present in Sx Modification Request to update the Traffic Endpoint information on the User Plane.

Following are the IEs in an Update Traffic Endpoint IE:

- Traffic Endpoint ID
- Local FTEID
- Network Instance
- UE IP address
- In addition to the 3GPP standards defined IEs, a private IE called "Bearer Info IE", is added to the Create Traffic Endpoint which includes:
- QCI of the bearer
- ARP of the bearer
- Charging ID of the bearer

NOTE: Currently, the Update Traffic Endpoint IE supports only the update of Private IE extensions, such as the Bearer Info IE. There are no use-cases wherein update of other information, such as Local FTEID, Network Instance, UE IP address, is required.

When the QCI/ARP of a particular bearer EPS-Bearer Identity (EBI) is modified, then the modified QCI/ARP along with the Charging ID is communicated to the User Plane with the help of Update Traffic Endpoint ID. A given Traffic Endpoint ID can be updated only if it was successfully created on the User Plane.

Remove Traffic Endpoint IE

This IE is present in Sx Modification Request to remove a traffic endpoint. Traffic Endpoint ID is included in the Remove Traffic Endpoint IE. A given Traffic Endpoint ID can be removed only if it is successfully created on the User Plane.

For Pure-S, Pure-P, and Collapsed call, when a bearer is deleted on the Control Plane, the Traffic Endpoints that are associated with the bearer are removed with Remove Traffic Endpoints. There is no explicit requirement to send Remove PDRs and Remove FARs on that bearer.

On the User Plane, for a Pure-S call, Remove Traffic Endpoints deletes all the PDRs, FARs, and URRs of that bearer. For Pure-P and Collapsed call, Remove Traffic Endpoints deletes all the PDRs, FARs, QERs, and URRs of that bearer.

PDI Changes in Create PDR

When PDI Optimization is enabled for the PDN, then the Traffic Endpoint ID is set in the PDI field of all PDRs of the bearers of the PDN. The PDI fields, such as F-TEID, PDN Instance, UE IP address, and so on, are not supposed to be filled and so, these fields are validated in the User Plane and error messages are posted in case of any validation failures. This is applicable for all interfaces, such as Sxa, Sxb, Sxab, N4, and Sxc.

PDI Optimization Changes on User Plane

Handling of Create Traffic Endpoint

When a Create Traffic Endpoint is received, the contents of the IE are validated for correctness. If validation fails, then an error message is sent back to the Control Plane.

Validations fail in the following cases:

- Basic IE validation failures.
- Traffic Endpoint exists with this Traffic Endpoint ID.
- CH-bit not set in the F-TEID IE inside Traffic Endpoint.
- PDN Instance is not valid.
- UE IP address is not valid.

When a Create Traffic Endpoint is successfully processed, then a local F-TEID is allocated by the User Plane and it is associated with the Traffic Endpoint. The Created Traffic Endpoint is sent back to Control Plane for this Traffic Endpoint with the F-TEID information and Traffic Endpoint ID.

When a Create Traffic Endpoint list is processed on the User Plane in Sx Establishment Request, PDI optimization is enabled for the lifetime of the Sx Session which cannot be changed midway.

Handling of Update Traffic Endpoint

When an Update Traffic Endpoint is received, the contents of the IE are validated for correctness. If validation fails, then an error message is sent back to the Control Plane.

Validations fail in the following cases:

- Basic IE validation failures.

- Traffic Endpoint with its Traffic Endpoint ID does not exist.

NOTE: Currently, Update Traffic Endpoint updates only bearer information, such as QCI, ARP, and Charging ID on the User Plane. Update is not supported for any other Traffic Endpoint parameters.

Handling of Remove Traffic Endpoint

When a Remove Traffic Endpoint is received, the contents of the IE are validated for correctness. If validation fails, then an error message is sent back to the Control Plane.

Validations fail in the following cases:

- Basic IE validation failures.
- Traffic Endpoint with its Traffic Endpoint ID does not exist.

When a Remove Traffic Endpoint is received, the PDRs associated with the Traffic Endpoint, FARs associated with the PDR, QERs associated with the PDR, and URRs associated with PDR are also removed.

To remove a bearer, the Control Plane sends Remove Traffic Endpoints for the Traffic Endpoints that are associated with the bearer resulting in the cleanup of the bearer-associated data on the User Plane.

The Control Plane does not explicitly send any Remove PDRs, Remove FARS, Remove QERS, or Remove URRs for a bearer removal. However, if the Control Plane does send Remove PDRs, Remove FARS, Remove QERS, or Remove URRs with Remove Traffic Endpoints, the message is accepted and successfully processed.

Handling of Create PDR

When Sx Session has the PDI Optimization enabled, the Traffic Endpoint ID is set for Create PDR. If not, an error response is sent back to the Control Plane. The Create PDR validation fails in the following cases:

- Basic IE validation failures.
- Create PDR does not have Traffic Endpoint ID set in the PDI IE.
- Create PDR has valid F-TEID IE in PDI IE.
- Create PDR has valid PDN Instance IE in PDI IE.
- Create PDR has valid UE IP address IE in PDI IE.

For a Sx Session with PDI optimization disabled, the Create PDR is validated for various other fields. If Traffic Endpoint ID is valid in PDI, then an error response is sent back to the Control Plane as Traffic Endpoint ID should not be present for a Sx Session with the PDI optimization being disabled.

Session Recovery and ICSR

Control Plane

Session Recovery and ICSR are supported for the Traffic Endpoint IDs of all bearers of a PDN. The Traffic Endpoint IDs are recovered for all bearers of a given PDN. This support is provided for Pure-S, Pure-P, and Collapsed call. With this, PDI optimization enabled status for a PDN is also recovered. Full Checkpoint is used for check-pointing and recovery of the Traffic Endpoints IDs of bearers.

User Plane

Session Recovery and ICSR are supported for the Traffic Endpoints on the User Plane for all bearers. All the Traffic Endpoints, that are associated with a given Sx Session, are recovered. For a given Traffic Endpoint, the associated PDR list is also recovered. For a given PDR, the associated Traffic Endpoint ID is recovered.

Standards Compliance

The PDI Optimization feature complies with the following standard: 3GPP TS 29.244 V15.5.0 (Interface between the Control Plane and the User Plane Nodes).

Limitations

The PDI Optimization feature has the following limitations:

- The Network instance and UE IP address IEs are currently not supported for a Pure-S call.
- The Update Traffic Endpoint IE supports only the update of Private IE extensions, such as the Bearer Info IE. Update of other information, such as Local F-TEID, Network Instance, UE IP address, are not supported.
- The Update Traffic Endpoint updates only bearer information, such as QCI, ARP, and Charging ID on the User Plane. Update is not supported for any other Traffic Endpoint parameters.

Configuring the PDI Optimization Feature

This section describes how to configure the PDI Optimization feature.

Enabling PDI Optimization

Use the following CLI commands to enable the feature.

```
configure
  context context_name
    sx-service service_name
      [ no ] sx-protocol pdi-optimization
      end
```

NOTES:

- **no**: Disables PDI optimization.
- By default, the CLI command is disabled.
- PDI Optimization is enabled or disabled at PDN level. PDI Optimization is enabled for each PDN based on the configuration in sx-service. The PDN is PDI Optimization-enabled if the configuration is enabled while processing Sx Establishment Request on the Control Plane.
- Configuration changes will not have any effect on the PDN. The configuration that is applied while processing Sx Establishment Request will be maintained throughout the lifetime of the PDN. In a multi-PDN call, each PDN has the configuration applied while PDN is set up.

- On the User Plane, there is no separate configuration to determine whether the PDN has PDI Optimization-enabled. When Create Traffic Endpoint IE is received in Sx Establishment Request for a Sx session, then the Sx session is considered to have PDI Optimization-enabled throughout the lifetime of the session. This will not change dynamically midway, and validations are done accordingly. In case of any validation failures, Error Response is sent back to the Control Plane.
- When there are multiple Create Traffic Endpoint IEs with the same Traffic Endpoint ID, the first Create Traffic Endpoint IE is processed, and rest are ignored. The same behavior is applicable for Created Traffic Endpoint IE, Update Traffic Endpoint IE, and Remove Traffic Endpoint IE.

Verifying the PDI Optimization Feature Configuration

To verify if the PDI Optimization feature is enabled or disabled, use the **show sx-service all** CLI command. The output of this show command has been enhanced to display the following:

- SX PDI Optimisation: [Enabled/Disabled]

PDI Optimization OAM Support

This section describes operations, administration, and maintenance information for this feature.

Show Command Support

The following show CLI commands are available in support of PDI Optimization feature.

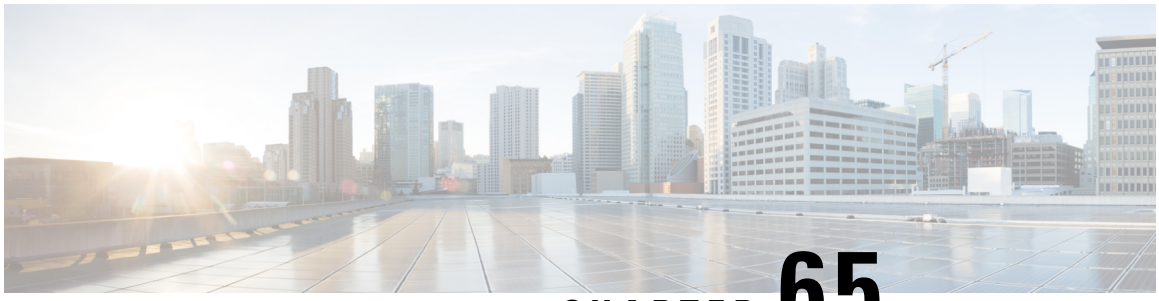
show subscribers user-plane-only callid <call_id> pdr all

The output of this CLI command has been enhanced to display the following field: Associated Create Traffic Endpoint-ID(s)

show subscribers user-plane-only callid <call_id> pdr full all

The output of this CLI command has been enhanced to display the following field:

- Create Traffic Endpoint-ID
  - Bearer QOS
    - QCI
    - ARP
    - Charging Id



CHAPTER 65

P-GW CDR in CUPS

- Revision History, on page 643
- Feature Description, on page 643
- User Location Information in P-GW CDR, on page 644

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

In CUPS architecture, support is added for P-GW CDR generation for custom24 GTPP dictionary. The P-GW CDR is generated for following procedure or scenario:

- Default Bearer:
  - Volume/Time Limit
  - PCRF initiated Rule Base change
  - S-GW/PLMN change due to S1 Handover
  - ULI/Time Zone change
  - QoS change
  - UE/Network initiated session deletion
  - RAN-NAS cause code
  - Maximum change condition trigger

- Dedicated Bearer:
  - Volume/Time Limit
  - QoS change
  - Handover Procedure
  - ULI/Time Zone change
  - PCRF rule base change
  - UE/Network initiated Dedicated Bearer Deletion Procedure
  - RAN-NAS cause code

Limitations

The aFRecordInformation is not supported in CUPS architecture.

User Location Information in P-GW CDR

The P-GW CDR contains the User Location Information (ULI) in the following two attribute fields:

- User Location Information (32)
- User Location Information (34-0-20)

As per the current behavior above two fields contain the “User Location information” in P-GW CDR. These fields are getting updated only when ULI-change trigger is enabled. If ULI-change trigger is not configured, the P-GW CDRs keeps the user location as it was reported in the initial CDR, even after the “Radio Access Technology” gets changed.

To overcome this issue, this feature was introduced, that even if “ULI-change trigger” is disabled, Every CDR contains the latest “User Location Information”. Functionality overview of this feature is as follows:

- This feature allows the P-GW CDRs to update User Location Information (32) and User Location Information (34-0-20) attributes with the latest User Location Information provided by the MME and S-GW.
- The implementation of the feature is through the different filler function specific to feature.
- To use this feature, customer/user requires to make the software changes at two places. First one is to update the CDR custom/customer’s dictionary ULI fields with the newly implemented filler functions. Current implementation is in the custom dictionary 38, as per requirement. Parallelly, the support for the same dictionary need to be added under the MACRO: “ACS_CHK_DICT_SUPPORT_FOR_LATEST_ULI”.

If the dictionary with the new filler functions are used, it packs the latest ULI in case of the following events:

Events to send/generate partial PGW-CDR for a subscriber:

- When the number of QoS changes or tariff time changes reaches the configured maximum number of charging condition changes.

- Before this, service containers are added to the CDR for every change.
- Every x seconds configured using "interval x".
- Every x octets configured using "volume x" (up/down/total).
- Command gtpp interim now active-charging egcdr.
- Transferring the context to a new S-GW/SGSN (serving Node Change).
- Changing the access type within the same P-GW (RAT Change).

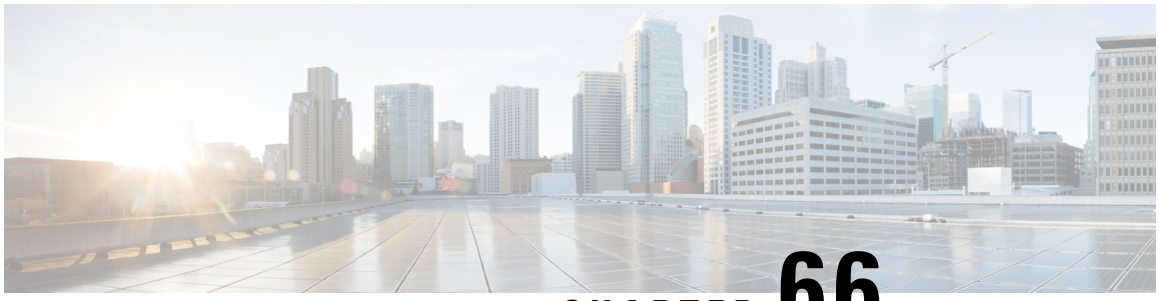
Events to send or generate the final P-GW CDR for a subscriber:

- Detach Request received from UE
- Delete bearer context request received from S-GW.
- Manual subscriber clearing
- Abnormal Releases such as path failures.

Sample Configuration

Following are the sample configurations:

```
Customer dictionary: custom38
Customer running configuration:
    gtpp group pgwhdd
      gtpp attribute local-record-sequence-number
      gtpp attribute node-id-suffix PGW11
      no gtpp attribute twanuli
      gtpp dictionary custom38
      no gtpp trigger dcca
      no gtpp trigger service-idle-out
      no gtpp trigger serving-node-change-limit
      no gtpp trigger inter-plmn-sgsn-change
      no gtpp trigger qos-change
      no gtpp trigger ms-timezone-change
      gtpp trigger egcdr max-losdv
      no gtpp trigger uli-change
      gtpp egcdr lotdv-max-containers 1
      gtpp egcdr losdv-max-containers 1
      gtpp suppress-cdrs zero-volume-and-duration gcdrs egcdrs
      gtpp egcdr service-data-flow threshold interval 43200
     gtpp egcdr service-data-flow threshold volume total 104857600
     gtpp storage-server mode local
gtpp storage-server local file purge-processed-files file-name-pattern

            ACQ* purge-interval 2880
      gtpp storage-server local file format custom3
      gtpp storage-server local file rotation volume mb 30
      gtpp storage-server local file rotation cdr-count 65000
      gtpp storage-server local file rotation time-interval 600
      gtpp storage-server local file name prefix PGW11_Laca
    #exit.
```



CHAPTER 66

P-GW Restart Notification

- Revision History, on page 647
- Feature Description, on page 647

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

P-GW Restart notification (PRN) procedure is supported for UP communication over the Sx interface during P-GW path failure. The P-GW Restart Notification procedure optimizes the amount of signaling involved on the S11/S4 interface when a P-GW failure is detected.

PRN procedure is a standards-based procedure supported on S-GW to notify detection of P-GW failure to MME/S4-SGSN.

P-GW failure detection will be done at S-GW when it detects that the P-GW has restarted (based on restart counter received from the restarted P-GW) or when it detects that P-GW has failed but not restarted (based on path failure detection).

When an S-GW detects that a peer P-GW has restarted, it locally deletes all PDN connection and bearer contexts associated with the failed P-GW and notifies the MME through P-GW Restart Notification.

The S-GW, in the echo request/response on S11/S4 interface, indicates that the P-GW Restart Notification procedure is supported.

P-GW Restart Notification Procedure is an optional procedure and is invoked only if both the peers, MME/S4-SGSN and S-GW, support it.

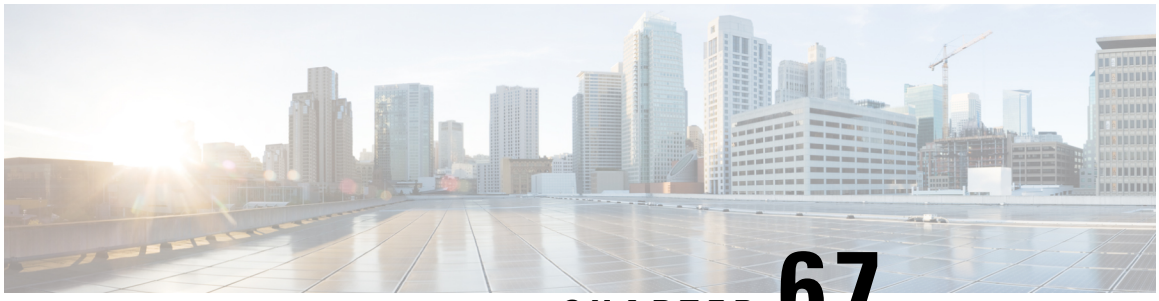
In the absence of this procedure, S-GW will initiate the Delete procedure to clean up all the PDNs anchored at that failed P-GW, which can lead to flooding of GTP messages on S11/S4 if there are multiple PDNs using that S-GW and P-GW.

The following figure illustrates the PRN flow during a path failure.

In CUPS, when a path failure is detected:

IMAGE HERE

- On detecting S5 pathfailure S-GW initiates PRN processing if S-GW and MME supports the PRN feature.
- For a path failed session, if S-GW has not sent a PRN message to MME then it will send PRN message once per MME.
- For path failed session, the S-GW CP sends a Sx Modify with FAR Action = DROP.
- On receiving Sx Modify Response, the S-GW CP sends Sx Delete Request to UP.



CHAPTER 67

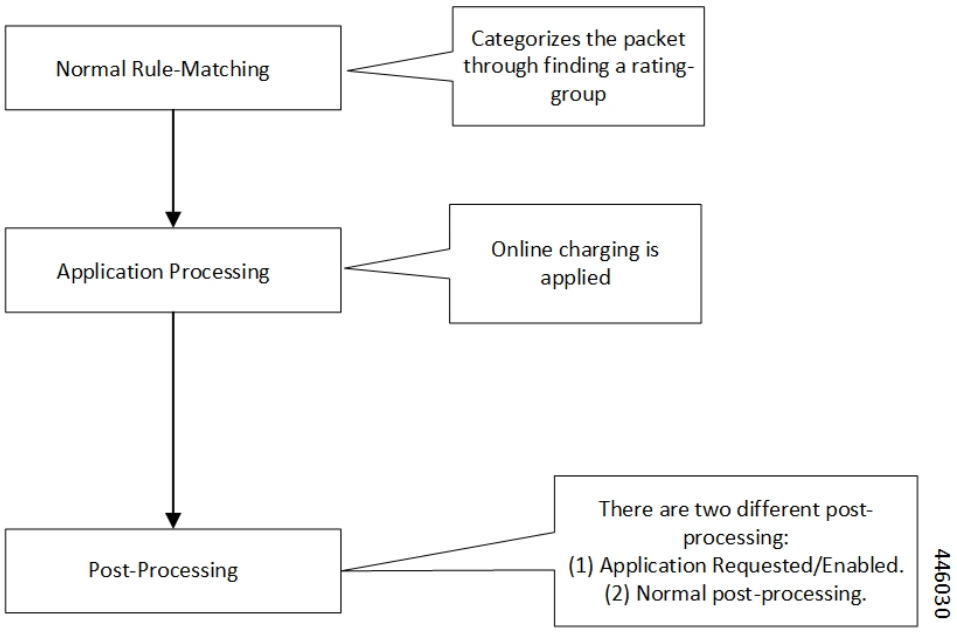
Post Processing Interaction for DCCA

- Feature Description, on page 649

Feature Description

The following diagram explains about the packet processing.

Figure 41: Post Processing Interaction for DCCA



Normal Rule Matching

In this phase, A comparison happens between the incoming packet against the configured rules in the box. This rule matching process is nothing but categorizing the packet. Use the following CLIs for the Rule Matching configuration in the box.

```
action priority <priority-number> ruledef <ruledef-name>
charging-action <charging-action>
```

Based on priority order, the Rule Matching happens against the packet. The first rule that matches categorizes the packet.

The corresponding charging-action applies to the packet. If the charging-action configuration contains “cca charging credit”, then it triggers the online charging, for which the packet moves to the DCCA application.

Application Processing

Once the packet reaches the DCCA Application, it checks the quota for the packet (rating-group/content-id) and makes the necessary processing. When there are no more credits for that rating-group, the Final-Unit-Actions takes place on the packet. If no-credit is present in that rating-group, DCCA can also blacklist the rating-group. When the application is blacklisting, the packet gets marked for Discard/Drop. The packet is in the disposition-action to inform the ACS mgr. If the quota is present, the packet goes for forwarding. The DCCA application can alternatively populate the post processing rules/filter list and mark the packet for postprocessing. The postprocessing happens when the OCS has requested for applying the filter-ids or filter-rules along with the Final-Unit-Indication AVPs. Once the DCCA application processing completes on the packet, it goes back to the ACS mgr.

Post Processing

When the packet returns from the application, the ACS MGR, sees the disposition action value set by the DCCA Application. If it’s marked for discard, it gets discarded.

- Application Requested Post-Processing: If the disposition-action applies for PP_RESTRICTION_RULE or PP_FILTER_ID, it tries to get the corresponding restrict-rules-list or restrict-filter-id-list for the content-id/rating-group. It applies the postprocessing. The packet doesn’t attempt for the below-post-processing (General Post-Processing).
  - ACS_CONTROL_PP_RESTRICTION_RULE: This disposition action applies, when the DCCA activates Restriction-Filter-Rules sent by OCS, inside the Final-Unit-Indication Grouped-AVP, as per RFC 4006. The Restriction-Filter-Rules are applicable in “restriction_list”, inside the “fui_restrict_access”.
  - ACS_CONTROL_PP_FILTER_ID: This disposition action applies, when the DCCA activates the Filter-Ids, the OCS inside the Final-Unit-Indication grouped-AVP, as per RFC4006. The Filter-Ids are nothing but the rule def names, and are applicable in “filter_id_list”, inside the “fui_restrict_access”

    DCCA Application can set both the disposition actions. Disposition-action is nothing but a bitmask.

    These postprocessing restrict rules or postprocessing filter ids, that came from OCS and enabled/activated by DCCA Application. This rule is rating-group specific rules. The rule-matches happen in the order in which the OCS sends.

    For each acs_sub_sess, there’s a list of “dcca_mscc_fui_restrict_access_t”, indexed on “service_id & rating_group”. For each of this combination, the preceding type structure exists. This “dcca_mscc_fui_restrict_access_t” structure contains the “filter_id_list” & “fui_restrict_access” lists. This structure gets empty by default. The DCCA application can fill it when it activates the corresponding post processing filtering for that service-id + rating-group.

- General Post Processing: If it’s forward, the post processing starts. During the post processing, the packet gets matched against the configured post processing rules in the boxer.

  Configure the post processing rules in boxer using the following CLIs:

Post processing priority *<priority-number> ruledef* **<ruledef-name> charging-action** *<charging-action-name>*

These post processing rules get matched against the packet in the order of the priority-number.

Limit Reached Post Processing

In addition to the preceding two disposition action values, there's one more value for limit-reached scenarios, it's ACS_CONTROL_PP_LIMIT_REACHED. Here the limit-reached indicates that the user quota-limit is over. When the user quota is over, the packets get dropped by default, by application, and no post processing applies. The feature is to add control on this limit-reached scenario, where post processing configuration happens, even for this quota exhausted scenario.

A configurable option is available for enabling the post processing for limit-reached/quota-exhausted packets. Use the following CLI for this configuration:

```
configure
  active-charging service service_name
    rulebase rulebase_name
      post-processing policy { always | not-for-dynamic-discard }
      end
```

The option "not-for-dynamic-discard" is the default option. This option indicates that the post processing doesn't apply for the limit-reached/quota-exhausted scenarios.

In case of the "post processing policy always" CLI, the post processing rules applies for the limit-reached/quota-exhausted scenarios. The "ACS_CONTROL_PP_LIMIT_REACHED" value in the disposition action is to communicate about this behavior. If there are post processing priority-based rules, it checks for any redirection rules, else discards the packets by default. No other post processing actions like forward, next-hop, X-header-insertion applies on these limit-reached packets.

Configuring Post Processing

The post processing rule def with the limit-reached case have "cca qutoa-state = limit-reached" configured, along with the "rule-application post processing" option. This configuration is to indicate that this rule def is for the limit-reached scenario.

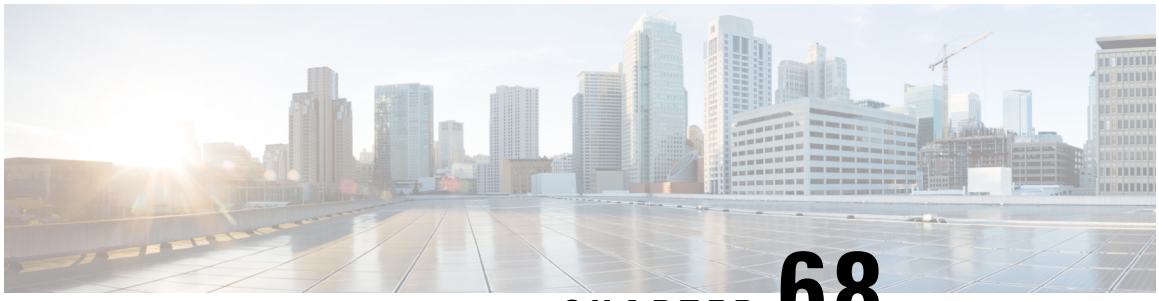
```
ruledef http_low
  http any-match = TRUE
  cca quota-state = limit-reached
  rule-application postprocessing
#exit
```

The corresponding charging-action has the "flow action redirect "configuration. Any other flow action values are invalid for the limit-reached scenario.

```
charging-action redirect
      flow action redirect-url http://webpages/index.html
    #exit
```

Configure the post processing priority rules in the rule base in such a way that the limit reached post processing rules is of the high priority. So that the packets get matched first against the limit-reached rule def.

```
rulebase base1
      ..................................
      post processing priority 1 ruledef http_low charging-action redirect
    #exit
```



CHAPTER 68

Priority Recovery Support for VoLTE Calls

- Feature Summary and Revision History, on page 653
- Feature Description, on page 653
- How It Works, on page 653
- Call Flows, on page 655
- Configuration, on page 656
- Monitoring and Troubleshooting, on page 657
- Show Commands and Outputs, on page 657

Feature Summary and Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

This feature helps to priorities the active and nonactive VoLTE calls over the normal calls. The priority is for the recovery of calls due to the failure of the User Plane.

Relationship

This feature is related to *VoLTE Support in CUPS*.

How It Works

There are two types of sessions in the User Plane:

- Normal Session

- Prioritized session

Prioritized session - The MP (message priority) bit set in PFCP header received from the Control Plane during the Sx Session establishment/modification request. The prioritized sessions take precedence in case of recovery. Normal calls recover only after the completion of the recovery of the prioritized calls.

The Control Plane sets the message priority (upper nibble of the 16th octet) in the PFCP header along with the MP (second bit of the first Octet). Currently for EMPS calls, Message Priority is 1. Similarly, message priority is 2 for VoLTE active calls and Message priority is 3 for VoLTE nonactive calls. Following figure describes the message priority in PFCP header format for the various calls.

	Bits							
Octets	8	7	6	5	4	3	2	1
1	Version			Spare	Spare	Spare	MP = 1	S=1
2	Message Type							
3	Message Length (1st Octet)							
4	Message Length (2nd Octet)							
5	Session Endpoint Identifier (1st Octet)							
6	Session Endpoint Identifier (2nd Octet)							
7	Session Endpoint Identifier (3rd Octet)							
8	Session Endpoint Identifier (4th Octet)							
9	Session Endpoint Identifier (5th Octet)							
10	Session Endpoint Identifier (6th Octet)							
11	Session Endpoint Identifier (7th Octet)							
12	Session Endpoint Identifier (8th Octet)							
13	Sequence Number (1st Octet)							
14	Sequence Number (2nd Octet)							
15	Sequence Number (3rd Octet)							
16	Message Priority = 1 EMPS/EMERGENCY = 2 for VoLTE active call = 3 for VoLTE nonactive						Spare	

On receipt of SX Session establish/modification request, the User Plane marks the session as prioritized session. The priority is based on nonzero (EMPS=1, VoLTE Active=2, VoLTE nonactive =3) value of the message priority filled in the PFCP header.

This feature supports the following aspects for the Priority Recovery of VoLTE calls.

On Control Plane: (P-GW, S-GW, SAE-GW, GGSN)

- VoLTE call configuration under APN

- Sets the MP priority Bit and Message Priority in the PFCP header of SX session establishment request
- Sets MP priority Bit and Message Priority in the PFCP header of SX session modification request

On User Plane:

- Checks the Message Priority of the PFCP header for the earlier messages
- If the message priority is nonzero, mark the session as priority session.
- These prioritized sessions are recovered before the nonprioritized sessions after SR /ICSR.

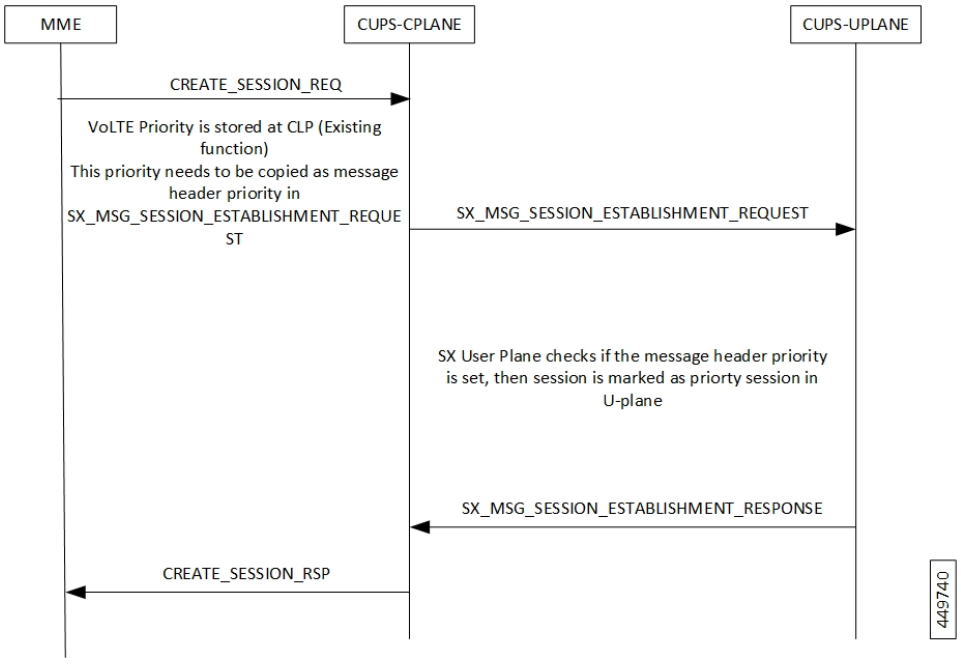
Call Flows

The following call flows explain about the:

- Session Establishment Handling
- Session Modification Handling

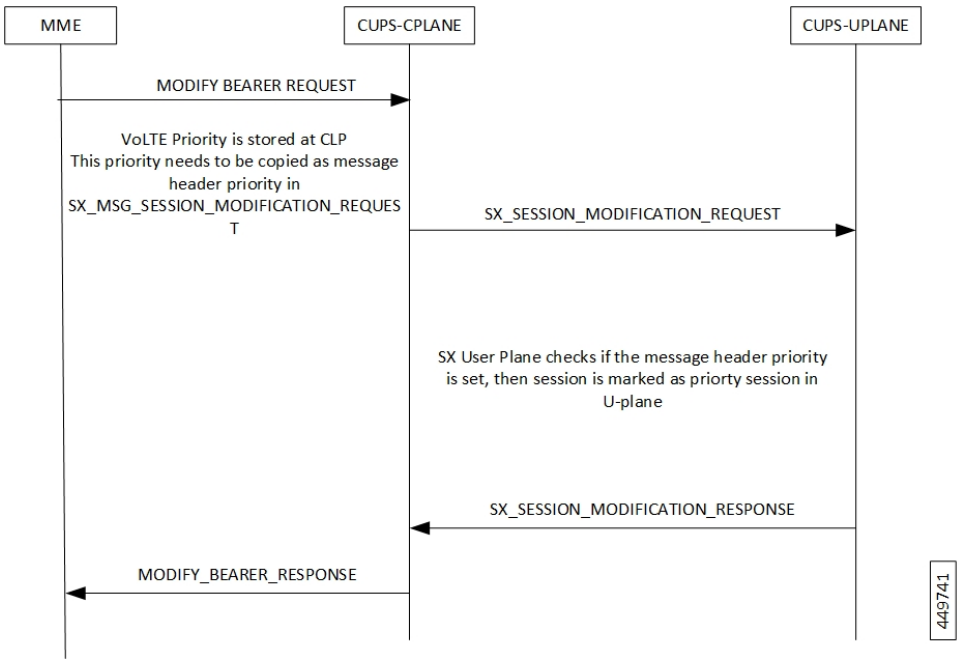
Session Establishment Handling Call Flow

The following call flow explains about the Session Establishment.



Session Modification Handling Call Flow

The following call flow explains about the Session Modification.



Configuration

Following are the configurations for the Pure-P/Collapsed calls and Pure-S calls.

Configuring Pure-P or Collapsed Calls

Following are the configurations to mark the calls as VoLTE in Control Plane for Pure-P/Collapsed calls.

```
configure
          context ingress
            apn vrf.com
            qci1 ims-media
            end
```

Configuring Pure-S Calls

Following are the configurations to mark the calls as VoLTE in Control Plane for Pure-S/Collapsed calls.

```
configure
          apn profile apn_1
          qci1 ims-media
configure
   operator-policy name intershat
   apn default-apn-profile apn_1
   end
configure
   lte-policy
   subscriber-map map_name
     precedence 1 match-criteria all operator-policy-name intershat
   end
```

```
configure
  context ingress
  sgw-service sa_sgw_service
  associate subscriber-map map_name
  end
```

Monitoring and Troubleshooting

This section provides information on CLI commands that are available for monitoring and troubleshooting for priority recovery of VoLTE calls.

Show Commands and Outputs

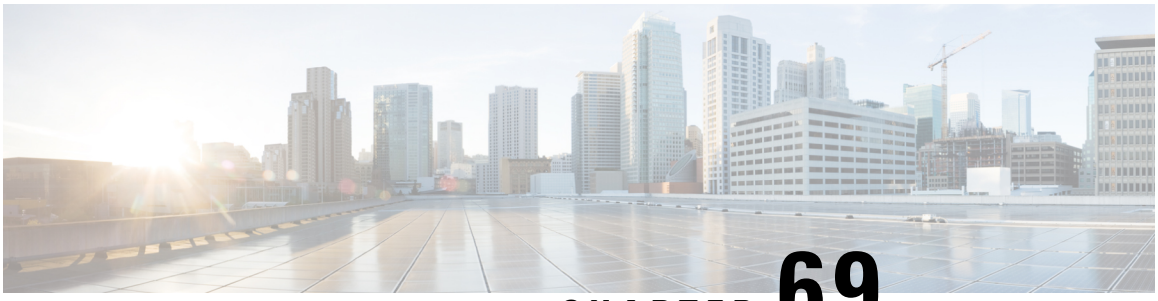
This section provides information about show CLI commands that are available in support of priority recovery of VoLTE calls in User Plane.

show session subsystem facility sessmgr instance 1 debug-info

```
AAA TCP Connect Succeeded with   : 0    Retries
fetched_from_aaamgr                      : 1    pror_to_audit                    : 1

passed_audit                             : 1    calls_recovered                  : 1

calls_recovered_by_tmr                   : 1    calls_recovered_by_med           : 0

priority_calls_recoverd_by_med           : 0    non_priority_calls_ignored_by_med: 0
```

show session subsystem facility aaamgr instance 1 debug-info

```
1 Current recovery archives 1 Current valid recovery records
1 Current valid priority recovery records
```



CHAPTER 69

QoS Group of Ruledefs Support

- Revision History, on page 659
- Feature Descriptions, on page 659
- How It Works , on page 659
- Monitoring and Troubleshooting, on page 662

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Descriptions

QoS Group of Ruledefs is also called as QGR or SGQ. This feature enables fair usage policing for the subscriber.

How It Works

The following configuration primarily does Flow-Status and Bandwidth Limiting in hierarchical manner, first doing at matched Charging-Action and then at QoS-Group Level.

```
conf
active-charging service acs
  qos-group-of-ruledefs QGR1
      add-group-of-ruledef group
      add-ruledef http
    #exit
rulebase cisco
action priority 2 ruledef http charging-action standard
action priority 5 ruledef catchall charging-action standard
```

```
route priority 1 ruledef http-rule analyzer http
end
```

QoS Group QGR1 received over PCRF.

```
qos-group-rule-install
qgr-name QGR2
qgr-mon-key 1
qgr-flow-status 3
qgr-precedence 1
qgr-eqos-information
qgr-eqos-mbr 1000 2000
qgr-eqos-mbr-burst-size 1000 2000
qgr-eqos-mbr-limit-conform-action 1 -1 1 -1
qgr-eqos-mbr-limit-exceed-action 2 7 2 8
```

Data Path Enforcement

1. Packet matches ruledef ‘http’.
2. QGR match is carried out to check if there is a QGR with the matched ruledef/group. Highest Priority QGR is returned. The ruledef/group can be static or predefined.
3. If QGR matches, then Flow-Action Enforcement which is first done at Charging-Action Level and then at QGR Level assuming Charging-Action has allowed the packet. If the packet is dropped, then QGR Level Flow Action Enforcement is skipped.
4. If Flow-Action at QGR allows the packet, then QER Limiting is enforced on a packet. If it is dropped at QGR, QER Limiting is skipped.
5. Likewise, QER Limiting is done stepwise, first at Charging-Action Level and then at the QGR subject to packet is allowed at Charging-Action.

Static Configuration Push to UPlane

- Static configuration pushed from CP to UP via the PFD mechanism in similar to ECS elements ruledef/charging-action/group-of-ruledefs.
- Show CLIs ‘show user-plane-service qos-group-of-ruledefs all/name’ displays the static configuration on UPlane.

QGR Params Push to UPlane

QGR is pushed along with Session Establishment and Modification Request.

QGR Name and Precedence is sent in a private IE. Flow-action, bandwidth parameters, and monitoring-key will create a new FAR, new QER, and new URR respectively.

Any changes to QGR dynamic parameters triggers an update to FAR/QER/URR.

This is sent in Session Establishment or Modification Request.

Private IE

```
Qos-Group-Of-Ruledef:
Name:
Operation:  (0 - Add  1 - Modify 2 - Delete)
```

```
Precedence:
FAR ID:
URR ID:
QER ID:
```

Table 45: FAR Format

FAR ID	Unique ID
Extended Apply Action	Private IE to include Flow-Action Allow as well Discard, Uplink, Discard Downlink, Terminate Flow.

Table 46: QER Format

QER ID	Unique ID
Maximum Bitrate	MBR of QGR in Kbps: UL MBR: DL MBR:
Burst Size	Private IE to include the Burst Size: UL Burst: DL Burst:
Conform Action	Private IE to configure the conform action: Uplink Action: Uplink ToS: Downlink Action: Downlink ToS:
Exceed Action	Private IE to configure the exceed action: Uplink Action: Uplink ToS: Downlink Action: Downlink ToS:

Display the FAR, PDR, QER, and URR in ‘show subscribers user-plane-only callid <> far|qer full all’.

Processing of QGR on UPlane

- On Receiving a IE ‘Qos-Group-Of-Ruledef', search for the QGR in static configuration. For each ruledef/group-of-ruledef in QGR, look up for its corresponding PDR and update the FAR/QER list with the received QGR FAR/URR/QER IDs.
- For each ruledef/group-of-ruledef PDR on UPlane, associate high priority QGR’s FAR-id, QER-id.

- Maintain QGR map at both Control and UPlane, it consists of QGR name, precedence, QER-ID, and FAR-ID. Use QGR map for recovery and lookup whenever required.

QGR Hit in Data Path

- For a packet matching rule PDR, search for the highest priority QGR FAR, and QER and enforce the parameters.
- Enforce flow-status and flow-rate as expected.
- QGR matching for Offloaded Flows are handled.
- QGR hit statistics are incremented.

Limitations

The QoS Group of Ruledefs support feature has the following limitations:

- URR creation and enforcement is not supported.
- Inclusion of dynamic-rules in static QGR definition is not supported.
- Flow-Status Redirect and Kill Flow are not supported.
- QoS Group Conform action as Drop and Exceed action as ALLOW or MARK_DSCP are not supported.
- CP can communicate maximum 20 QGRs received over PCRF to UP.

Monitoring and Troubleshooting

This section provides information about CLI commands available for monitoring and troubleshooting the feature.

Show Commands and Outputs

This section provides information about show commands and their outputs in support of this feature.

show subscribers user-plane-only full all

The output of this show command has been enhanced to include the following fields introduced in support of this feature.

- Total QoS-Group Active
- QoS-Group Statistics
  - QGR Name
  - Pkts-Down
  - Bytes-Down

- Pkts-Up
- Bytes-Up
- Hits
- Match-Bypassed
- FP-Down(Pkts/Bytes)
- FP-Up(Pkts/Bytes)

show user-plane-service qos-group-of-ruledefs all name

The output of this show command has been enhanced to include the following fields introduced in support of this feature.

QGR-INFO-LIST

- Value
- Number of QGRs
- QGR INFO
  - NAME
  - PRECEDENCE
  - OPERATION
  - FAR ID
  - QER ID
- QGR INFO
  - NAME
  - PRECEDENCE
  - OPERATION
  - FAR ID
  - QER ID

show subscribers user-plane-only callid 00004e21 qos-group all

The output of this show command has been enhanced to include the following fields introduced in support of this feature.

```
Callid: 00004e21
           Interface Type: Sxb
        QGR-Name:      Priority:       FAR-ID:         QER-ID:         URR-ID:
        ---------      ---------       -------         -------         ------


       Total Number of QGRs found:
```

show subscribers user-plane-only callid 00004e21 far full all

The output of this show command has been enhanced to include the following fields introduced in support of this feature.

- Associated with QGR
  - Extended Apply Action

show subscribers user-plane-only callid 00004e21 qer full all

The output of this show command has been enhanced to include the following fields introduced in support of this feature.

- UL Burst
- UL Conform Action
  - UL DSCP Value
- UL Exceed Action
  - UL DSCP Value
- DL Burst
- DL Conform Action
  - DL DSCP Value
- DL Exceed Action
  - DL DSCP Value

show subscribers user-plane-only callid 00004e21 qos-group statistics all name

This show command and its output is introduced to support of this feature.

- Flow-Status Statistics
  - Total Uplink Packets
  - Total Uplink Bytes
  - Uplink Packets Redirected
  - Uplink Bytes Redirected
  - Uplink Packets Dropped
  - Uplink Bytes Dropped
  - Uplink Packets Term-Flow
  - Uplink Bytess Term-Flow
  - Total Downlink Packets

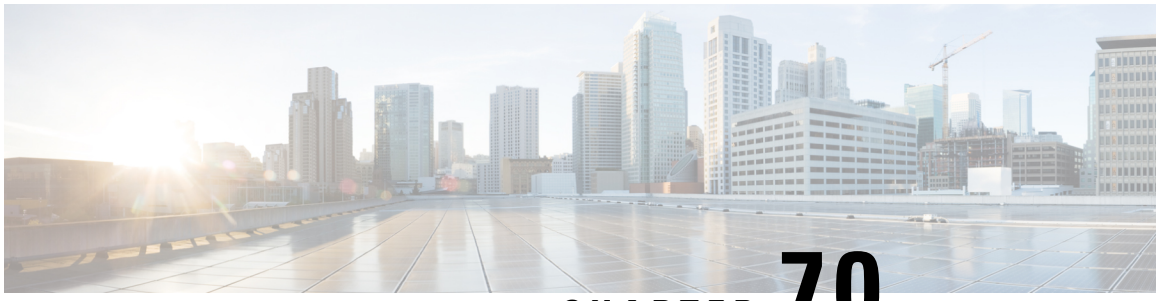
    - Total Downlink Bytes
    - Downlink Packets Redirected
    - Downlink Bytes Redirected
    - Downlink Packets Dropped
    - Downlink Bytes Dropped
    - Downlink Packets Term-Flow
    - Downlink Bytes Term-Flow
- Bandwidth-Control Statistics
    - Total Uplink Packets
    - Total Uplink Bytes
    - Uplink Packets QoS-Exceed
    - Uplink Bytes QoS-Exceed
    - Uplink Packets QoS-Conform
    - Uplink Bytes QoS-Conform
    - Uplink Packets Dropped
    - Uplink Bytes Dropped
    - Uplink Packets Marked
    - Uplink Bytes Marked
    - Total Downlink Packets
    - Total Downlink Bytes
    - Downlink Packets QoS-Exceed
    - Downlink Bytes QoS-Exceed
    - Downlink Packets QoS-Conform
    - Downlink Bytes QoS-Conform
    - Downlink Packets Dropped
    - Downlink Bytes Dropped
    - Downlink Packets Marked
    - Downlink Bytes Marked
- Total qos-group-of-ruledefs matched
- Total subscribers matching specified criteria

show user-plane-service statistics qos-group sessmgr all

Sessmgr Instance

- Total Uplink Pkt
- Total Uplink Bytes
- Uplink FP Pkts
- Uplink FP Bytes
- Total Dnlink Pkts
- Total Dnlink Bytes
- Dnlink FP Pkts
- Dnlink FP Bytes
- Flow-Status Statistics
  - Total Uplink Packets
  - Total Uplink Bytes
  - Uplink Packets Redirected
  - Uplink Bytes Redirected
  - Uplink Packets Dropped
  - Uplink Bytes Dropped
  - Uplink Packets Term-Flow
  - Uplink Bytess Term-Flow
  - Total Downlink Packets
  - Total Downlink Bytes
  - Downlink Packets Redirected
  - Downlink Bytes Redirected
  - Downlink Packets Dropped
  - Downlink Bytes Dropped
  - Downlink Packets Term-Flow
  - Downlink Bytes Term-Flow
- Bandwidth-Control Statistics
  - Total Uplink Packets
  - Total Uplink Bytes
  - Uplink Packets QoS-Exceed
  - Uplink Bytes QoS-Exceed

- Uplink Packets QoS-Conform
- Uplink Bytes QoS-Conform
- Uplink Packets Dropped
- Uplink Bytes Dropped
- Uplink Packets Marked
- Uplink Bytes Marked
- Total Downlink Packets
- Total Downlink Bytes
- Downlink Packets QoS-Exceed
- Downlink Bytes QoS-Exceed
- Downlink Packets QoS-Conform
- Downlink Bytes QoS-Conform
- Downlink Packets Dropped
- Downlink Bytes Dropped
- Downlink Packets Marked
- Downlink Bytes Marked



CHAPTER 70

Rate Limiting Function (RLF)

This chapter contains the following topics:

- Revision History, on page 669
- Feature Description, on page 669

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The RLF feature implements a generic framework that can be used by multiple interfaces and products for rate-limiting/throttling outgoing messages like Diameter messages on Gx, Gy interface towards PCRF.

Important The working of RLF feature, including the CLI commands, in the CUPS architecture is similar to how it works in the non-CUPS environment.

When applications send messages to peers at a high rate (for example, when a large number of sessions goes down at the same time), accounting stop messages for all the sessions are generated at the same time) the peer may not be able to handle the messages at such high rates. To overcome this situation, the Rate Limiting Function (RLF) framework is developed so that the application sends messages at an optimal rate such that peer is capable of receiving all the messages and does not enter an overload condition.

This feature can be enabled using the **rlf-template** CLI command in the Global Configuration mode. The users can define the rate limiting configurations within this template. For more information on the command, see the *Command Line Interface Reference*.

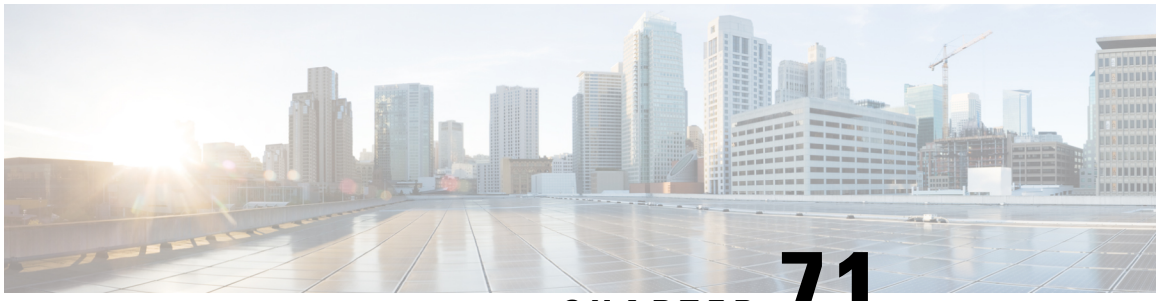
Important RLF template cannot be deleted if it is bound to any application (peers/endpoints).

When RLF feature is enabled, all the messages from the application are pushed to the RLF module for throttling and rate control, and depending on the message-rate configured the RLF module sends the messages to the peer. Once the rate or a threshold value is reached, the RLF module notifies the application to slow down or stop sending messages. RLF module also notifies the application when it is capable of accepting more messages to be sent to the peer. RLF module typically uses a Token Bucket Algorithm to achieve rate limiting.

Currently in the deployment of the Diameter applications (Gx, Gy, and so on), many operators make use of **max-outstanding** *number* CLI command as a means of achieving some rate-limiting on the outgoing control traffic. With RLF in place, this is no longer required since RLF takes care of rate-limiting in all cases. If both RLF and **max-outstanding** is used, there might be undesirable results.

Important If RLF is being used with a **diameter endpoint**, then set the **max-outstanding** value of the peer to be 255.

To use the template, Diameter or any other applications must be associated with the template. The RLF provides only the framework to perform the rate limiting at the configured Transactions Per Second (TPS). The applications (like Diameter) should perform the configuration specific to each application.



CHAPTER 71

S2a Interface Support

- Revision History, on page 671
- Feature Description, on page 671

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

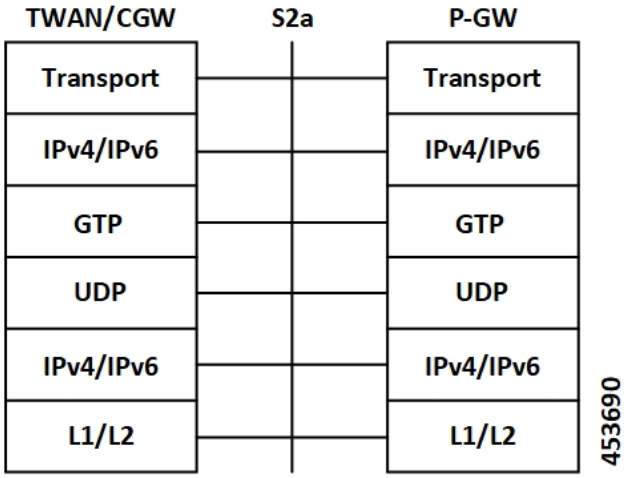
Feature Description

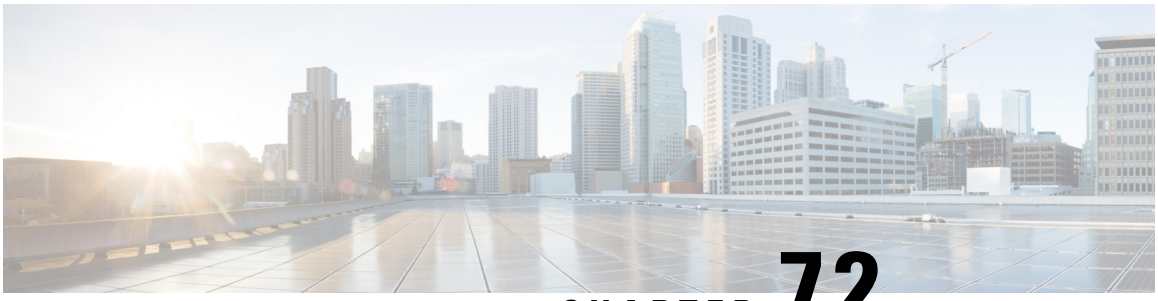
This reference point supports the bearer interface by providing signaling and mobility support between a trusted non-3GPP access point (Trusted WiFi Gateway (TWAN)/Converged Access Gateway (CGW)) and PDN Gateway (P-GW). It is a GTP based interface support that allows the connectivity to the trusted non-3GPP IP access points. The S2a interface uses IPv4 and IPv6 for both control and data.

Supported Protocols

- Transport Layer: UDP, TCP
- Tunneling: GTP IPv6
- Network Layer: IPv4, IPv6
- Data Link Layer: ARP
- Physical Layer: Ethernet

Figure 42: Protocols Supported on the S2a Interface





CHAPTER 72

S2b Interface Support

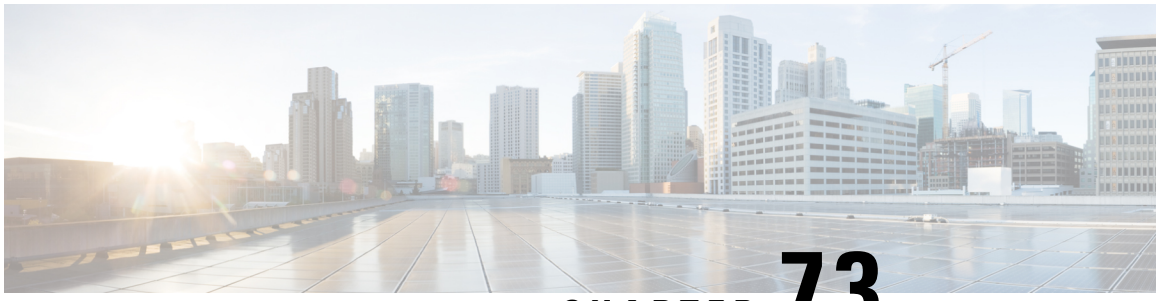
- Feature Description, on page 673

Feature Description

In CUPS architecture, support is added for S2b interface where untrusted Wi-Fi calls from ePDG connects to SAEGW (Pure-P).

Currently, support for following procedures are available:

- Support procedures for session establishment:
  - GTP based S2b for roaming, non-roaming and LBO (3GPP TS 23.402 [4] clause 7.2.4).
  - Emergency services over GTP based S2b (3GPP TS 23.402 [4] clause 7.2.5).
  - UE-initiated connectivity to additional PDN from Un-trusted Non-3GPP IP Access with GTP (3GPP TS 23.402 [4] clause 7.6.3).
- Support procedures for session release:
  - UE/ePDG-initiated detach procedure with GTP on S2b (TS 23.402 [4] clause 7.4.3.1).
  - HSS/AAA-initiated detach procedure with GTP on S2b (TS 23.402 [4] clause 7.4.4.1).
- Support procedure for bearer deactivation:
  - P-GW Initiated Bearer Deactivation with GTP on S2b (TS 23.402 [4] clause 7.9.2).



CHAPTER 73

S-GW CDR in CUPS

- Revision History, on page 675
- Feature Description, on page 675

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

CDR generation is supported for S-GW in the Cisco UPC CUPS architecture.

CDRs in CUPS is generated to collect charging information for UE bearers in S-GW. On receiving a charging trigger, the Control Plane node of CUPS pulls the information from the corresponding user plane nodes and the collected volume counts are added to the S-GW CDR.

S-GW CDR is supported for both default and dedicated bearer.

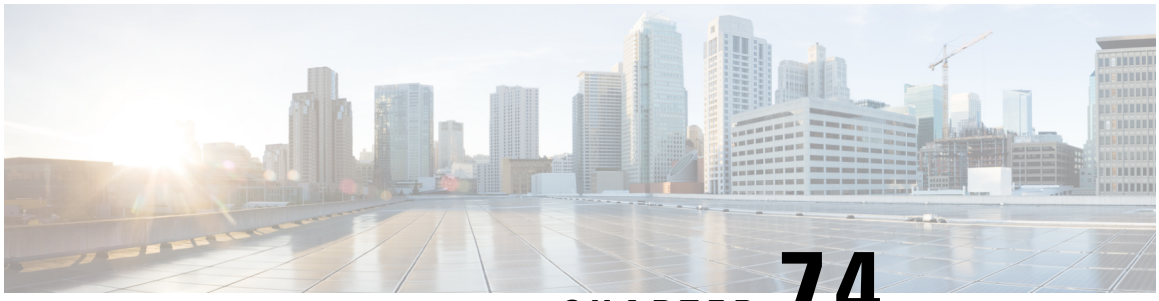
Note Currently, S-GW CDR is supported in custom24 dictionary.

Charging data is collected based on the following triggers:

- Access-side triggers:
  - ULI Change
  - RAT Change
  - Management intervention (Interim CDRs are not supported)

- Normal/Abnormal call release

- Network-side triggers:
  - QCI Change
  - APN AMBR Change



CHAPTER 74

S-GW New Call Rejection

- Feature Description, on page 677
- How It Works, on page 677
- Configuring S-GW New Call Rejection, on page 678
- Monitoring and Troubleshooting, on page 679

Feature Description

This CLI-controlled feature allows to reject Pure-S calls based on subscriber type (Roamer, Homer, Visitor), and/or APN.

Note This feature is applicable only when CUPS is enabled.

How It Works

When a new call arrives at S-GW, and if the feature CLI is enabled with which the APN of the call matches to the one configured through the CLI, the call is rejected. This feature works by identification of the type of subscribers—homer, visitor, or roamer. This identification is done in the following way:

- If the PLMN ID of S-GW is same as that of PGW and International Mobile Subscriber Identity (IMSI), the subscriber is identified as homer.
- If the PLMN ID of S-GW differs from PLMN ID of PGW irrespective of IMSI, the subscriber is identified as roamer. For example, if MS-1 is subscribed to PLMN1 and is connected to an SGW in PLMN2, then from PLMN2, MS-1 initiates a session with the PGW in PLMN1. In this scenario, MS-1 is roamer.
- Subscribers whose IMSI contains a foreign PLMN ID are identified as visitors.

The S-GW rejects all sessions of APNs that are configured for home, visitor, or roamer subscriber. Initial attach CS Request and UE requested additional PDN connection CS requests for Pure-S calls are also considered for rejection. The CS request is rejected with GTPV2 cause *No Resource Available*. The expected behaviour is that the MME reattempts attach based on this cause code, and blacklist this S-GW based on its blacklist algorithm implementation.

A configuration for list of APNs (maximum 10), which needs to be rejected by S-GW for homer and roamer subscribers, is required.

In case of SAEGW deployment, only Pure-S calls are rejected. If SAEGW receives CS request for collapsed call, then this call is not rejected even if corresponding APN is configured in the reject list.

Emergency or eMPS calls are not rejected, despite IMS APN being configured for new call reject, when:

- The S-GW receives CS request with IMS APN and unauthenticated imsi flag set.
- The S-GW receives CS request with IMS APN and eARP value is configured as eMPS eARP in S-GW service.

Note In the CS Request, eARP is received by S-GW, which is not configured as eMPS eARP. While in CS Response, the S-GW can receive new authorized eARP which can mark sessions as eMPS session. However, if the feature is enabled in case of CS Response, sessions are rejected while handling CS Request only.

Limitations

When Pure S call is rejected through new call reject policy, the rejection statistics is collected under *New Call Policy Rejection Stats* section of the **show saegw-service statistics all function sgw** CLI command. Other SGW-related statistics for the rejected call are not collected.

Configuring S-GW New Call Rejection

This section provides information on configuration commands to enable and disable support for S-GW to reject new calls.

Enabling New Call Rejection

Use the following configuration commands to reject calls at S-GW for roamer, home, visitor subscribers, and APN subscribers.

```
configure
   context context_name
      sgw-service sgw-service_name
         [ default | no ] newcall reject { roamer  | home [ apn apn_name ]
| visitor [ apn apn_name }
         end
```

NOTES:

- **default**: Resets the command to it its default setting - Disabled.
- **no**: Disables the rejection of all calls for the specified subscriber.
- **newcall**: Configures a new call for the configured S-GW service.
- **reject**: Configures newcall reject-policy for the configured S-GW service home, visitor, or roamer subscriber.

- **roamer**: Configures newcall reject-policy for the configured S-GW service for roamer subscriber.
- **home**: Configure newcall reject-policy for the configured S-GW service for home subscriber.
- **visitor**: Configures newcall reject-policy for the configured S-GW service for visitor subscriber.
- **apn-name** *apn_name*: Configures the APN name (for maximum of 10 APN profiles) to reject call for the configured S-GW service for home or visitor subscriber.

Monitoring and Troubleshooting

This section provides information regarding commands available to monitor and troubleshoot the new call and APN session rejection at S-GW.

Show Command(s) and/or Outputs

This section provides information about show commands and the fields that are introduced in support of new call and APN session rejection at S-GW.

show saegw-service statistics all function sgw

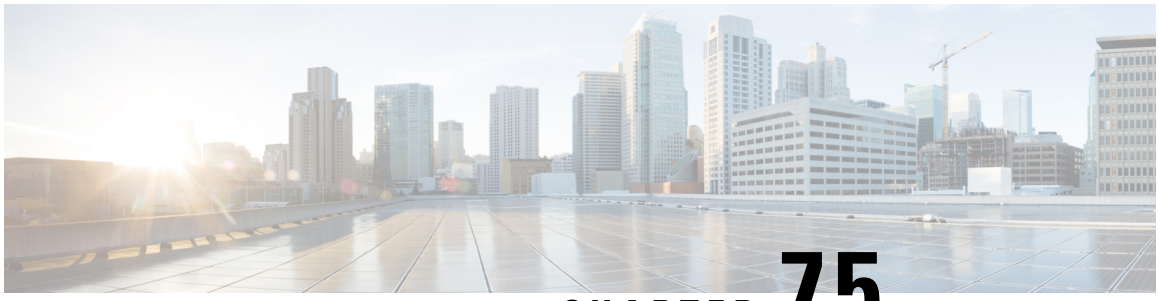
The output of this show command has been modified to display apn-profiles that are configured in sgw-service for new call rejection. Following fields are introduced:

- New Call Policy Rejection Stats
- New Calls
  - Visiting Subscriber
  - Home Subscriber
  - Roaming Subscriber

show sgw-service name

The output of this show command has been modified to display apn-profiles that are configured in sgw-service for new call rejection. Following fields have been introduced:

- SGW Reject Calls Visitor Subs
- SGW Reject Calls Roamer Subs
- SGW Reject Calls Home Subs



CHAPTER 75

S-GW Session Idle Timeout

- Revision History, on page 681
- Feature Description, on page 681
- Configuring Session Idle Timeout, on page 682

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

This chapter describes the Idle Timeout Handling feature for S-GW sessions. On the ASR5500 platform, subscriber session is represented by call-line. The S-GW product call-line interfaces to its peers through MME/S4-SGSN on S11/S4 and P-GW on S5/S8. In some scenarios, peer sessions are deleted by respective peers, S-GW does not receive or miss deletion messages, and as a result S-GW session remains idle. Such idle or stale sessions are counted towards valid call-lines in system for effectively consuming resources and causing capacity reduction. In such cases, S-GW triggers to get the new subscriber session, which results in the removal of old session for same subscriber. The Idle Timeout Handling support enables the identification of such sessions and initiates deletion to release the resources.

The following points describes the idle timeout handling for S-GW sessions:

- The subscriber session is idle when there is no data traffic activity for the subscriber. The session manager keeps track of the call-line state, when no data traffic is recorded for call-line, such sessions are moved to idle state.
- Session which is idle for defined timeframe referred as idle timeout is considered for idle timeout handling. In idle timeout session, S-GW initiates the deletion of session towards its peers.

- Idle timeout is configured in seconds depending on the network requirements. The timeout range is 1-4294967295 seconds.
- The idle timeout configuration is applicable on S-GW service level for enabling the idle timeout handling for set of subscribers handled by that service.

Configuring Session Idle Timeout

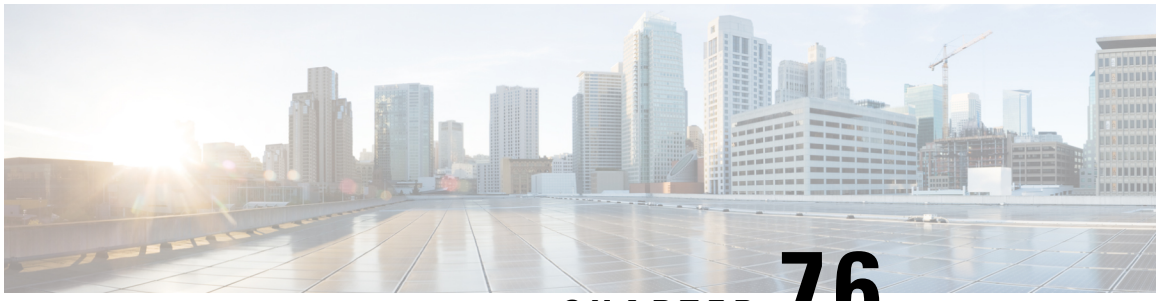
The session idle timer for S-GW sessions is configurable from S-GW service.

To configure Session Idle Timeout for S-GW, use the following configuration:

```
configure
   context context_name
      sgw-service service_name
         [ no | default] timeout idle timeout_duration
         end
```

NOTES:

- **timeout idle** *timeout_duration*: Specifies the maximum duration a session can remain idle for, in seconds, before the system automatically terminates the session. *timeout_duration* must be an integer in the range of 1-4294967295. 0 disables the feature. By default, it is disabled for the S-GW service.



CHAPTER 76

SAEGW Idle Buffering with DDN Delay and DDN Throttling

- Revision History, on page 683
- Feature Description, on page 683
- How It Works, on page 683
- SAEGW Idle Buffering with DDN Delay and DDN Throttling Support Configuration, on page 693

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The Downlink Data Notification (DDN) messages with support for DDN Delay and DDN Throttling, and buffering in SAEGW, when UE is in Idle State, is supported in CUPS architecture.

How It Works

This section provides an overview of how this feature works.

- Buffering is supported at SAEGW-U.
- Support of buffering starts when UE moves to IDLE state due to Release Access Bearer.
- ACTIVE to IDLE transition:

- When the UE moves to ECM-IDLE state, since the SAEGW supports buffering capability and decides to activate buffering in SAEGW-U for the session, the SAEGW-C informs the SAEGW-U through an Sx session modification.
- After the buffering starts, when the first downlink packet arrives on any bearer, the SAEGW-U informs the SAEGW-C. The SAEGW-U sends an Sx reporting message to the SAEGW-C, unless specified otherwise, and identifies the S5/S8 bearer on which the downlink packet is received.
- On receiving the reporting message, the SAEGW-C decides whether to send a DDN message to the MME, as defined in 3GPP TS 23.401 [2]. The DDN notification is sent with the Sx-Usage-Report.

- IDLE to ACTIVE transition:
  - At the UE transition to ECM-CONNECTED state, the SAEGW-C updates the SAEGW-U through Sxa interface with the F-TEIDu of the eNodeB/RNC/SGSN. The buffered data packets, if any, are then forwarded to the eNodeB/RNC/SGSN by the SAEGW-U.

- If the Apply Action is BUFFER, and SGW-U recovers, the SGW-U initiates Sx Report (with DLDR Report Type) on arrival of the downlink data packet.
- In SGW-U, a timer is implemented that starts after each Sx Report (with DLDR report Type) is sent. If the Apply Action is not changed then on timer expiry, Sx Report (with DLDR Report Type) gets initiated again.
- ARP of the bearer is included in the DDN message.
- In a multi-PDN session, if the DDN is initiated for one PDN and then data is received on another PDN, wherein the bearer has higher priority, then the DDN is initiated again with the higher priority ARP value.

Downlink Data Notification – Delay (DDN-D) Support

Under certain conditions, when UE triggers a service request, uplink and downlink data is triggered and is received at the SGW-C even before the Modify Bearer Request (MBR) is received causing unnecessary Downlink Packet Notification messages sent that increases the load in MME.

In such cases, the MME monitors the rate at which these events occur. If the rate becomes significant (as configured by the operator) and the MME's load exceeds an operator configured value, the MME indicates "Delay Downlink Packet Notification Request" with parameter D to the Serving Gateway, where D is the requested delay given as an integer with multiples of 50 milliseconds, or zero. The S-GW then uses this delay in between receiving downlink data and sending the Downlink Data Notification message.

The Downlink Data Notifications are supported for both Collapsed and Pure-S calls.

Due to the distributed nature of the system, sessions from a particular MME are offloaded on different session managers. Therefore, all session managers are notified when a session is offloaded. Also, the functionality is designed to not allow all session managers to message the DEMUX manager.

- In DDN Delay feature, DDN delay timer support is at Control Plane.
- When first data packet arrives, Sx Report message is initiated but DDN message is initiated from Control Plane after the expiry of Delay timer.
- DDN Delay feature is a peer level feature and so, it is applied for all the session on that peer from where the DDN Delay value is received.

- In case a previous delay value was received from a peer and it is absent in the current message, the delay value will be considered as 0.

Session Recovery and ICSR is supported for DDNs.

DDN Throttling Support

Too many DDN requests towards MME from SGW-C could lead to processing overload at MME. To reduce this load, MME dynamically requests SGW-C to reduce a certain percentage of DDN messages sent towards it for a given period time.

For DDN throttling, S-GW is required to drop a given percentage of DDNs over a given period of time. S-GW implements this functionality using a probabilistic algorithm at each session manager.

Whereas, the conventional implementation of DDN throttling requires each session manager to share its list of pending DDNs for low priority bearers with a central entity that would then calculate the net load of pending DDNs and then decide how many DDNs each session manager would have to drop. This implementation would require buffering of DDN messages at session manager. Also, due to distributed processing nature of software subsystem in chassis, it would require considerable amount of messaging between the session managers and the central entity (demuxmgr in case of Boxer) at regular intervals.

Implementing a probabilistic algorithm removes the need for buffering at session manager and also messaging with demuxmgr. Accuracy of probabilistic algorithm increase with increasing low ARP priority paging load at session manager. Even with lower paging load, accuracy would be fairly close to the throttling factor provided.

For non-release 10 compliant MME, SGW_C provides option to enable throttling through the CLI.

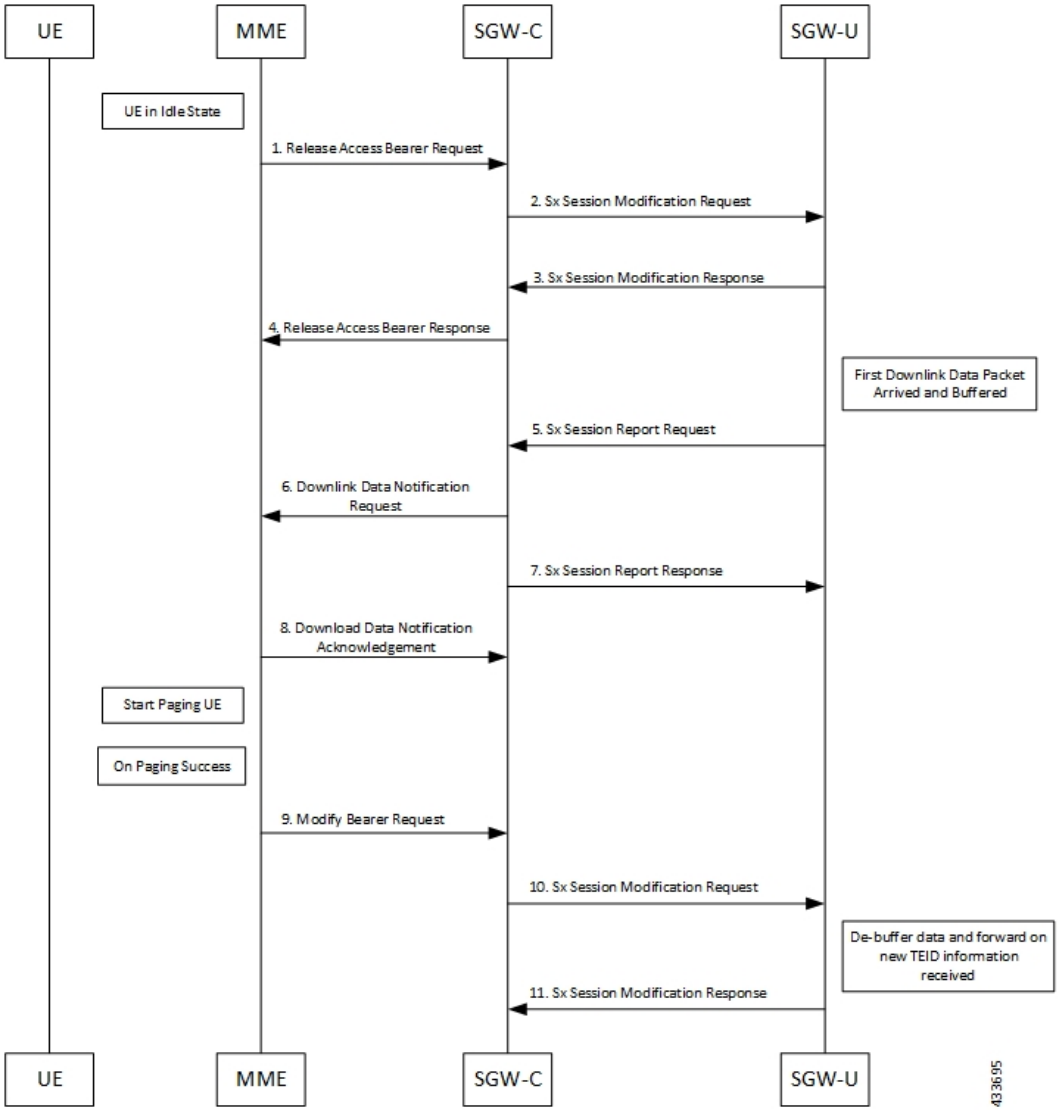
Threshold ARP values for low priority bearer must be configured through S-GW Service Configuration. For example, if configured ARP value is 9, any bearer with ARP > 9 is considered low priority bearer. DDN throttling is enabled through this configuration. If DDN throttling is enabled through SGW service configuration, each DDN message towards MME would contain the ARP IE.

No User Connect Timer Support

- Timer is introduced when a Modify Bearer Request is not received after positive Downlink Data Notification acknowledgment.
- It is initiated at SGW-C when DDN acknowledgment is received.
- On arrival of Modify Bearer Request, SGW-C stops this timer.
- On timer expiry SGW-C informs SGW-U to drop buffered packets.

DDN Call Flows

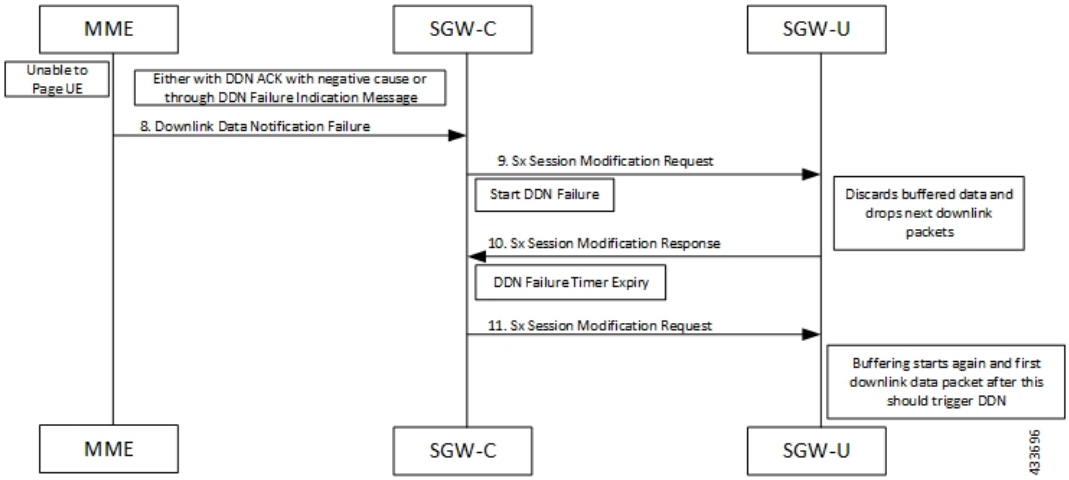
DDN Success Scenario



1. MME sends Release Access Bearer request to SGW-C to release downlink remote TEIDs of all the bearers for that UE.
2. On arrival of Release Access Bearer request, SGW-C informs the same to SGW-U by updating FAR with Apply Action as BUFFER in Sx Modification Request for all the PDNs.
3. SGW-U send Sx Modification response after applying Buffering in SGW-U for corresponding PDN.
4. SGW-C sends Release Access Bearer response to MME.
5. First Downlink data arriving in SGW-U triggers Sx Report Request (with Report Type as Downlink Data Report) towards SGW-C.

6. On arrival of Sx Report Request message, the SGW-C initiates Downlink Data Notification request message towards MME.
7. SGW-C sends Sx Report Response message towards SGW-U.
8. If MME is able to send a paging request towards UE, it sets the cause as “Request Accepted” in Downlink Data Notification Acknowledgment Message and sends it to SGW-C.
9. On successful paging, MME sends a Modify Bearer request to the S-GW with eNodeB TEIDs that sets up the S1-U connection at the SGW.
10. SGW-C sends Sx Modification request with updated FAR for new TEID information to SGW-U. SGW-U can now forward all the buffered data to UE through eNodeB.
11. SGW-U sends Sx Modification response to SGW-C.

DDN Failure Scenario



1. MME sends Release Access Bearer request to SGW-C to release downlink remote TEIDs of all the bearers for that UE.
2. On arrival of Release Access Bearer request, SGW-C informs the same to SGW-U by updating FAR with Apply Action as BUFFER in Sx Modification Request for all the PDNs.
3. SGW-U send Sx Modification response after applying Buffering in SGW-U for corresponding PDN.
4. SGW-C sends Release Access Bearer response to MME.
5. First Downlink data arriving in SGW-U triggers Sx Report Request (with Report Type as Downlink Data Report) towards SGW-C.
6. On arrival of Sx Report Request message, the SGW-C initiates Downlink Data Notification request message towards MME.
7. SGW-C sends Sx Report Response message towards SGW-U.
8. If MME is not able to page UE then it can reject Downlink Data Notification Request with relevant cause.

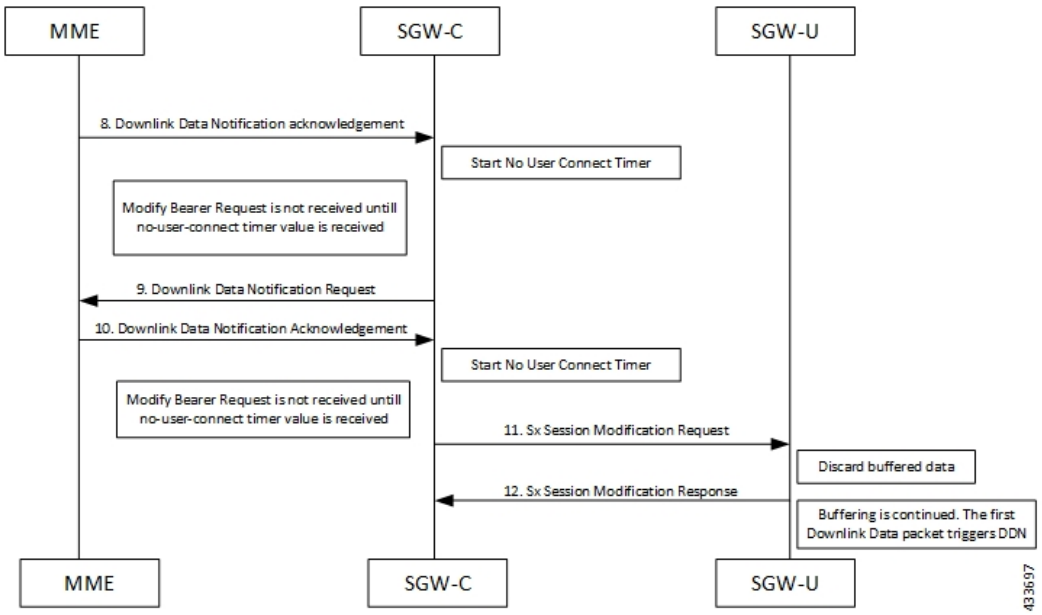
   OR

If MME accepts Downlink Data Notification Request. But later sends Downlink Data Notification Failure indication in order to indicate SGW-C that the UE did not respond to paging.

9. SGW-C received DDN failure and hence to stop sending next DDN immediately , SGW-C starts DDN Failure Timer.SGW-C sends Sx Modification Request with DROBU flag to discard buffered packets and Apply Action as DROP to drop subsequent packets.
10. SGW-U sends Sx Modification Response to SGW-C.
11. On DDN Failure Timer Expiry SGW-C initiates Sx Modification with Apply Action as BUFFER in order to start buffering again.

Further steps are continued from Step 3 in theDDN Success Scenario, on page 686 call flow.

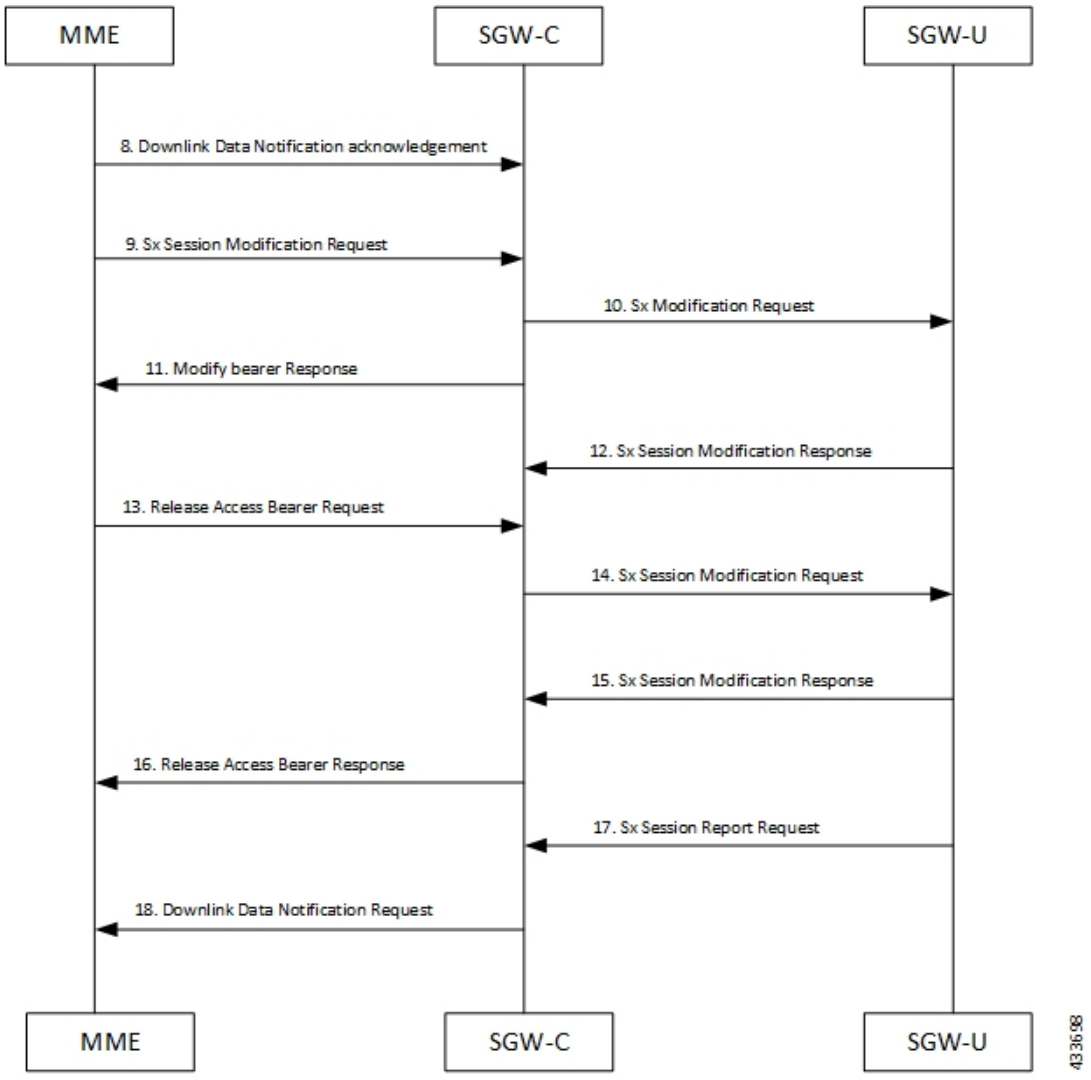
No User Connect Timer Support



1. MME sends Release Access Bearer request to SGW-C to release downlink remote TEIDs of all the bearers for that UE.
2. On arrival of Release Access Bearer request, SGW-C informs the same to SGW-U by updating FAR with Apply Action as BUFFER in Sx Modification Request for all the PDNs.
3. SGW-U send Sx Modification response after applying Buffering in SGW-U for corresponding PDN.
4. SGW-C sends Release Access Bearer response to MME.
5. First Downlink data arriving in SGW-U triggers Sx Report Request (with Report Type as Downlink Data Report) towards SGW-C.
6. On arrival of Sx Report Request message, the SGW-C initiates Downlink Data Notification request message towards MME.
7. SGW-C sends Sx Report Response message towards SGW-U.

8. Downlink Data Notification Acknowledgment is received from MME.SGW-C starts no-user-connect timer.
9. If the Modify Bearer request with eNodeB TEID information is not received and no-user-connect timer expires, SGW-C sends Downlink Data Notification again.
10. Downlink Data Notification Acknowledgment is received from MME. SGW-C initiates the no-user-connect timer again.
11. SGW-C initiates Sx Session Modification request towards SGW-U with DROBU flag set in the message. On receiving this flag SGW-U drops the buffered data. New data will be buffered, and the subsequent first packet initiates a Sx Report message for initiating Downlink Data Notification message.
12. SGW-U sends Sx Modification Response.

DDN Delay Timer



1. MME sends Release Access Bearer request to SGW-C to release downlink remote TEIDs of all the bearers for that UE.
2. On arrival of Release Access Bearer request, SGW-C informs the same to SGW-U by updating FAR with Apply Action as BUFFER in Sx Modification Request for all the PDNs.
3. SGW-U send Sx Modification response after applying Buffering in SGW-U for corresponding PDN.
4. SGW-C sends Release Access Bearer response to MME.
5. First Downlink data arriving in SGW-U triggers Sx Report Request (with Report Type as Downlink Data Report) towards SGW-C.
6. On arrival of Sx Report Request message, the SGW-C initiates Downlink Data Notification request message towards MME.
7. SGW-C sends Sx Report Response message towards SGW-U.
8. Downlink Data Notification Acknowledgment is received from MME with DDN Delay Timer value. This timer value will be saved for this peer , and now onwards every Downlink Data notification that we initiate should be after this delay for that peer.
9. On success paging, MME sends a Modify bearer request to the SGW with eNodeB TEIDs that sets up the S1-U connection at the SGW.
10. SGW-C sends Sx Modification Request with updated FAR for new TEID information to SGW-U. SGW-U can now forward all the buffered data to UE via eNodeB.
11. SGW-C sends Modify Bearer Response to MME.
12. SGW-U sends Sx Modification Response to SGW-C.
13. MME sends Release Access Bearer Request to SGW-C to release downlink remote TEIDs of all the bearers for that UE.
14. On arrival of Release Access Bearer Request, SGW-C inform the same to SGW-U via updating FAR with Apply Action as BUFFER in Sx Modification Request for all the PDNs.
15. SGW-U send Sx Modification Response after applying Buffering in SGW-U for corresponding PDN.
16. SGW-C sends Release Access Bearer Response to MME.
17. First Downlink data arriving in SGW-U triggers Sx Report Request (with Report Type as Downlink Data Report) towards SGW-C.
18. On arrival of Sx Report Request message, SGW-C starts DDN Delay Timer. On DDN Delay timer expiry SGW-C Initiates Downlink Data Notification message towards MME.

Sx Interface

Sx Session Level Reporting Procedure

Detection of first Downlink Data for Idle-Mode UE (by SAEGW-U):

When SAEGW-U receives the downlink packet but no S1-bearer for transmission and the buffering is performed by SAEGW-U, it reports the detection of first downlink data to SAEGW-C, for the purpose of paging the UE.

PFCP Session Report Request

The PFCP Session Report Request is sent over the Sxab interface by the User Plane function to report information related to a PFCP session to the Control Plane function.

Information elements	P	Condition / Comment	Appl.				IE Type
			Sxa	Sxb	Sxc	N4	
Report Type	M	This IE shall indicate the type of the report.	X	X	X	X	Report Type
Downlink Data Report	C	This IE shall be present if the Report Type indicates a Downlink Data Report.	X	-	-	X	Downlink Data Report

Downlink Data Report IE within PFCP Session Report Request

The Downlink Data Report grouped IE is encoded as shown in the following table.

Octet 1 and 2		Downlink Data Report IE Type = 83 (decimal)					
Octets 3 and 4		Length = n					
Information elements	P	Condition / Comment	Appl.				IE Type
			Sxa	Sxb	Sxc	N4	

PDR ID	M	This IE shall identify the PDR for which downlink data packets have been received at the UP function. More than one IE with this type may be included to represent multiple PDRs having received downlink data packets.	X	-	-	X	PDR ID

Notification to User Plane Function for DDN Failure

The Control Plane function notifies User Plane function for any failure so that buffered packets can be dropped and DDN related flags can be reset through DROBU flag in PFCP Sx Modification message.

PFCPSMReq-Flags	C	DROBU (Drop Buffered Packets): The CP function shall set this flag if the UP function is requested to drop the packets currently buffered for this PFCP session (see NOTE 1).

Limitations

Following are the known limitations of this feature:

- SAEGW Buffering is done for five data packets per PDN session.
- DDN profile configuration is not supported.
- Support for buffered data (data packet stream) that get deleted due to Flow Idle Timeout or other cases, is not present.

SAEGW Idle Buffering with DDN Delay and DDN Throttling Support Configuration

DDN Throttling for Release 10 Compliant MME

DDN throttling is enabled through Call Control Profile by providing the ARP value. For example, if the ARP value provided is 10, then all bearers with ARP value between 10-15 are treated as low priority bearers and are given throttling treatment. Throttling would not be enabled if ARP value is not provided through S-GW service configuration. Also, ARP IE in DDN message towards MME would not be included unless DDN throttling is configured using S-GW service. If MME is Release 10 compliant, the user need not configure the duration value as the DDN Acknowledgment would have the throttling IE. Otherwise, throttling can be enabled at S-GW by setting the duration value. If it's set to 0, S-GW would apply throttling recurringly. To enable throttling only for a given duration of time (in non Rel-10 compliant MME), user needs to set the value in hours and minutes. From the time of configuration, throttling would be applied at S-GW until the timer duration expires. For example, if user sets hours = 10, minutes = 30, S-GW would apply throttling for next 10 hours 30 minutes.

On re-configuration, all the parameters will be set with new values, but they will be applicable only from the next recalibration except from polling time and time factor.

Use the following configuration to configure DDN throttling for release 10 MME:

```
configure
  context context_name
    sgw-service service_name
      [ no ] ddn throttle arp-watermark arp_value
      end
```

NOTES:

- **arp-value**: Valid ARP value between 1 and 15. All the packets which have ARP greater than the configured values will be throttled as per the throttling factor.

DDN Throttling for non-Release 10 Compliant MME

Use the following configuration to configure DDN throttling for a non-release 10 MME:

```
configure
  context context_name
    sgw-service service_name
      ddn throttle arp-watermark arp_value [ rate-limit limit time-factor
seconds throttle-factor percent increment-factor percent [ poll-interval seconds
 ] throttle-time-sec seconds [ throttle-time-min minutes ] [
throttle-time-hour hour ] stab-time-sec seconds [ stab-time-min minutes ] [
stab-time-hour hour ]
      no ddn throttle
      end
```

NOTES:

- **rate-limit**: DDN permitted per second.
- **time-factor**: Time period in seconds over which SGW makes throttling decision (valid range 1-300 seconds.
- **arp-value**: Valid ARP value between 1 and 15. All the packets which have arp greater than the configured values will be throttled as per the throttling factor.
- **throttling-factor**: Percentage of DDN to be dropped upon detecting DDN surge (valid range between 1-100).
- **throttling-time-sec**: Time period in seconds over which DDN are throttled at SGW (valid range between 0-59 seconds).
- **throttling-time-min**: Time period in minutes over which DDN are throttled at SGW (valid range between 0-59 minutes).
- **throttling-time-hour**: Time period in hours over which DDN are throttled at SGW (valid range between 0-310 hours).
- **increment-factor**: Percentage value by which throttling factor is incremented dynamically, if existing throttling factor is insufficient to curb the DDN surge.
- **poll-interval**: Time in seconds (optional argument, default value = 1 second, poll interval < time-factor)
- **stab-time-sec/min/hours**: Stabilization time factor, time period over which if DDN rate returns to normal, then throttling need not be applied over entire throttling time period.

DDN throttling for non-Release-10 compliant MME makes use of existing Release-10 throttling implementation at SGW. By providing a configuration mechanism for SGW service, operator can still apply ddn throttling without needing any feedback from MME. Some salient points of this feature are described below:

1. The CLI configuration is applied per MME/S4-SGSN. Throttling parameters are tracked independently per MME/S4-SGSN.
2. On configuring this feature through CLI, demuxmgr polls each sessmgr for number of DDNs sent. By default, polling is done every second. This time interval can be changed by configuring the poll-interval time. Greater the poll interval time, lesser the number of internal messages within the chassis. However, it would take longer to detect a DDN surge.
3. By configuring time-factor, operator can specify the time interval for S-GW to apply throttling, if needed. It allows for some surge of DDNs if the net DDN rate is within specified limit over time-factor time interval. For example, time-factor= 10 seconds, ddn rate = 1000, poll interval = 2 seconds. Demux would poll each sessmgr every 2 seconds. Acceptable DDN rate limit is 1000*10 = 10000 DDNs every 10 seconds. Say after 2 seconds, 4000 DDNs were sent, in that case S-GW wouldn't apply throttling till rate limit of 10000 DDNs is crossed within time period of 10 seconds. This allows for intermittent bursts of DDNs.
4. DDN rate limit is configured through CLI. For example, if DDN rate limit is 1000 and poll interval = 1 second, time-factor = 5 seconds, then acceptable rate limit is 5000 DDNs over 5 seconds. If the number of DDNs sent by S-GW is greater than 5000 after 5 seconds, demuxmgr would ask all sessmgrs to initiate throttling.
5. Percentage of DDNs to be throttled is configured through throttling-factor.

6. Operator can specify increment-factor to increment throttling factor if existing throttling factor is insufficient to curb the DDN surge. For example, if throttling-factor = 10%, ddn-rate = 1000, increment-factor=10%. Once throttling is applied, S-GW drops ~10% DDNs. However, if DDN rate is still greater than 1000, S-GW would increase throttling-factor to 20%. If this is still not sufficient, it would be incremented to 30%. After incrementing throttling factor, if number of DDNs dropped are greater than expected, throttling-factor would then be decrement by increment-factor. E.g. in this case, after increasing throttling factor to 30%, if DDNs sent is less than 1000 per second (taking time-factor and poll-interval into consideration), throttling factor would be decremented to 20. The cap for decrementing throttling-factor would be the configured value (10% in this case).
7. Operator can configure the time duration for which throttling is applicable at S-GW. This could be a large value in order of days (for example: 10 days or 240 hours). The operator has an option to stop throttling if DDN rate is well under control by configuring stabilization time factor. In such a case, DDNs won't be needlessly dropped. For example, throttling-time =10 days, stab-time = 8 hours. After S-GW starts DDN throttling, in a time span of 8 hours, DDNs sent + DDNs dropped < ddn-rate * 8 hours, throttling would be stopped.

Show Commands Input and/or Outputs

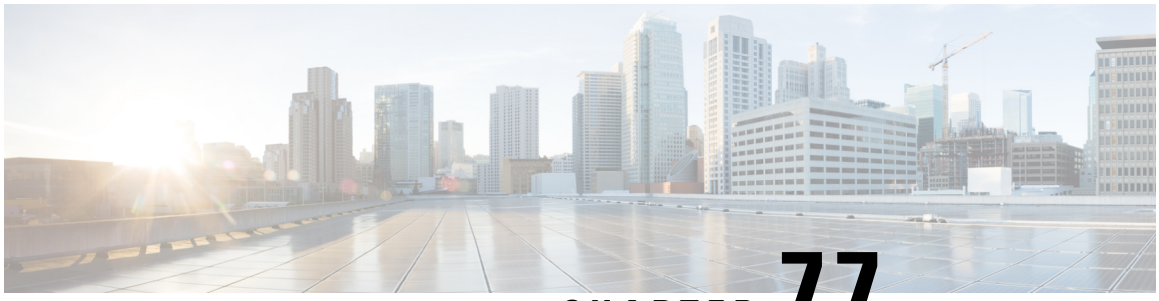
This section provides information regarding show commands and their outputs in support of the feature.

show subscribers user-plane-only-full all

The output of this command displays the following fields in support of this feature:

- buffered pkts
- buffered bytes
- buffer overflow drop pkts
- buffer overflow drop bytes

show subscribers user-plane-only-full all



CHAPTER 77

Secondary RAT Usage Report in CDR Records

- Revision History, on page 697
- Feature Description, on page 697
- Configuring Secondary RAT Usage Report through GTPP, on page 701
- Monitoring and Troubleshooting, on page 704

Revision History

Revision Details	Release
Support for "Secondary RAT Usage Report in CDR Records" feature has been added in this release as well.	21.20.31
Support for "Secondary RAT Usage Report in CDR Records" feature has been added in this release as well.	21.26
First introduced	21.23.14

Feature Description

Reporting issues pertaining to 5G **RANSecondaryRATUsageReport** occur due to lack of:

- Control in identifying whether the **RANSecondaryRATUsageReport** must be processed in CDRs or not. This allows the S-GW, P-GW, and SAEGW to either include these reports in the SGW-CDR or PGW- CDR or to simply ignore them.
- Number of available reports inside a CDR, if the control is active.
- Control in identifying whether Zero-volume reports must make it inside the CDR or not.

This results in billing loss of data. To overcome these reporting issues, you can trigger CLI controls using GTPP group configuration to:

- Allow the S-GW, P-GW, and SAEGW to either include the RANSecondary RAT Usage reports in the SGW-CDR or PGW-CDR or to simply ignore them.

- Identify the number of secondary RAT usage reports available inside the SGW-CDR or the PGW- CDR.

Note This limit must be in accordance with the system capability and ensure to consider the File-Format of the CDRs. If the configured limit exceeds, the system closes the SGW-CDR or PGW-CDR with the appropriate change-condition. For example, **max-change-condition** CDR is reused for further reports.

- Add or ignore Zero-volume reports inside the CDR.
- The CLI **gtpp limit-secondary-rat-usage** or hardcoded limit will be removed and the CLI **gtpp limit-secondary-rat-usage** is reused to control the number of records within the range 1-100.
- Provides logging when the CDR size reaches the maximum size. Through PGW-CDR counter, you can monitor the number of occurrences when the CDR exceeds its size limit.

Behavior Matrix

The following table explains the new behavior of P-GW and S-GW for this feature.

CLI	P-GW New Behavior	S-GW New Behavior
gtpp attribute secondary-rat-usage By default this CLI command is enabled in gtpp group.	P-GW sends secondary RAT usage records in CDR including zero volume records.	S-GW sends secondary RAT usage records in CDR including zero volume records.
[no] gtpp attribute secondary-rat-usage	P-GW does not send secondary RAT usage records in CDR.	S-GW does not send secondary RAT usage records in CDR.
gtpp suppress-secondary-rat-usage zero-volume By Default, this CLI command is disabled in gtpp group.	P-GW does not include and send zero volume secondary RAT records in CDR. P-GW sends only secondary RAT records that is having non-zero volumes.	S-GW does not include and send zero volume secondary RAT records in CDR. S-GW sends only secondary RAT records that is having non-zero volumes.
[no] gtpp suppress-secondary-rat-usage zero-volume	P-GW sends secondary RAT usage records including zero volume records in CDR.	S-GW sends secondary RAT usage records including zero volume records in CDR.

CLI	P-GW New Behavior	S-GW New Behavior
gtpp limit-secondary-rat-usage *range_1–100*. If not configured, the default value is 32. By default this CLI command is enabled in gtpp group. **Example: gtpp limit-secondary-rat-usage 32** **Note** This CLI is the modification of the existing CLI command **gtpp limit-secondary-rat-usage** with range between 1– 100.	PGW generates CDR immediately when total received secondary RAT records exceeds 32 and reported cause value is *maximum change condition*. P-GW generates multiple CDRs if total received secondary RAT records are multiples of 32. **Example**: If P-GW receives 100 RAT records between two triggers, PGW generates 3 CDRs and keeps the remaining 4 RAT records for the next CDR trigger.	S-GW generates CDR immediately when total received secondary RAT records exceeds 32 and the reported cause value is *maximum change condition*. S-GW generates multiple CDRs if the total received secondary RAT records are multiples of 32. **Example**: If S-GW receives 100 RAT records between two triggers, S-GW generates 3 CDRs and keeps the remaining 4 RAT records for the next CDR trigger.
Example:gtpp limit-secondary-rat-usage 40	P-GW generates CDR immediately when total received secondary RAT records exceeds 40 and cause value is *maximum change condition*. P-GW generates multiple CDRs if total received secondary RAT records are multiples of 40. **Example**: If P-GW receives 100 RAT records between two triggers, it will generate 2 CDRs and will keep remaining 20 RAT records for the next CDR trigger.	If the configured value is greater than 32 and sends 32 secondary RAT records in every CDR, Ignores **gtpp limit-secondary-rat-usage 40** CLI command.
Example:gtpp limit-secondary-rat-usage 20	P-GW generates CDR immediately when total received secondary RAT records exceed 20 and cause value is *maximum change condition*. P-GW generates multiple CDRs if total received secondary RAT records are multiples of 20. **Example:** If P-GW receives 100 RAT records between two triggers, P-GW generates 2 CDRs and will store the remaining 20 RAT records for next CDR trigger.	S-GW generates CDR immediately when the total received secondary RAT records exceeds 20 and cause value is *maximum change condition*. S-GW generates multiple CDRs if total received secondary RAT records are in multiples of 20. **Example**: If S-GW receives 100 RAT records between two triggers, it will generate 5 CDRs.

CLI	P-GW New Behavior	S-GW New Behavior
[no] gtpp limit-secondary-rat-usage	Generates CDR immediately when the total received secondary RAT records exceed 255 and cause value is *maximum change condition*. Generates multiple CDRs if the total received secondary RAT records are multiples of 255. **Example**: If 1000 RAT records between two triggers are received, then 3 CDRs are generated. The remaining 235 RAT records are stored for the next CDR trigger.	Ignores the **[no] gtpp limit-secondary-rat-usage** CLI and sends 32 secondary RAT records in every CDR. Behavior is similar to the **gtpp limit-secondary-rat-usage** *32* CLI implementation. Counter and debug logs are not required as it will never exceed the CDR size of 64k.
	Service specific Unit Limit is sending in **serviceConditionChange** file	Record Closure

Relationship to Other Features

- Sessmgr Restart While Processing Secondary RAT Usage CDR Records in the *P-GW Administration Guide*.
- Secondary RAT Usage IE during GnGp handover, S-GW, and P-GW support of Secondary RAT Data Usage Report in Gz CDRs, see the *5G Non-Standalone* chapter in the *P-GW Administration Guide*.
- P-GW support of Secondary RAT Data Usage Report in Rf CDRs, see the *5G Non-Standalone* chapter in the *P-GW Administration Guide*.

Limitations

This feature has the following limitations:

- Only 16 secondary RAT records per bearer are recovered for S-GW during session recovery. S-GW allows checkpointing of a maximum number of 16 secondary RAT records per bearer.
- A maximum of 142 secondary RAT records across all bearers is recovered for P-GW during session recovery. P-GW allows checkpointing of a maximum number of 142 secondary RAT records across all bearers.
- Anything beyond these numbers gets lost during session recovery.

Configuring Secondary RAT Usage Report through GTPP

Use the following GTPP configurations to close Secondary RAT Usage CDR records before exceeding a buffer size.

Enabling or Disabling the Secondary RAT Usage Report

Use the following configuration to enable or disable secondary RAT Usage report.

```
configure
  context context_name
    gtpp group group_name
      gtpp attribute secondary-rat-usage
      default gtpp attribute secondary-rat-usage
      no gtpp attribute secondary-rat-usage
      end
```

NOTES:

- **gtpp attribute secondary-rat-usage**: Sends an optional attribute Secondary RAT usage records.
- **default gtpp attribute secondary-rat-usage**: Sends an optional attribute Secondary RAT usage records by default.
- **no gtpp attribute secondary-rat-usage**: Does not send the optional attribute Secondary RAT usage records.

Controlling the Maximum Number of Entries

When the Secondary RAT usage record reaches the maximum configured value within a CDR, the CDR closure cause occurs and uses **maxChangeCond**. The **gtpp limit-secondary-RAT-usage** CLI command controls the maximum number of Secondary RAT usage record entries in the P-GW and S-GW CDRs. If the limit is configured with a value more than 32, the partial CDRs get generated with a maximum of 32 for S-GW CDR.

Note The existing behaviour of S-GW has a limit of 32 Secondary RAT Usage records.

The following table explains the behavior of Secondary RAT records and CDR, and the maximum limit.

Sl. Number	CDR Type	Configured limit-secondary-rat-usage	Effective Maximum Limit	No. of Secondary RAT records Sent by UE	Behavior of Secondary RAT Records and CDR
1	P-GW	Less than 32 Example: 20	20	35	Partial CDR is generated with 20 secondary RAT records.
					Remaining 15 secondary RAT records sent in the next trigger.
	S-GW	Less than 32 Example: 20	20	35	Partial CDR is generated with 20 Secondary RAT records.
					Remaining 15 Secondary RAT records sent in the next trigger.
2	P-GW	32	32	35	Partial CDR is generated with 32 Secondary RAT records.
					Remaining 3 secondary RAT records sent in the next trigger.
	S-GW	32	32	35	Partial CDR is generated with 32 secondary RAT records.
					Remaining 3 secondary RAT records sent in the next trigger.

Sl. Number	CDR Type	Configured limit-secondary-rat-usage	Effective Maximum Limit	No. of Secondary RAT records Sent by UE	Behavior of Secondary RAT Records and CDR
3	P-GW	Greater than 32 Example: 100	100	100	Partial CDR is generated with 100 secondary RAT records.
	S-GW	Greater than 32 Example: 100	32	100	Three partial CDRs are generated with 32 secondary RAT records each.
					Remaining 4 secondary RAT records sent in the next trigger.
4	P-GW	Not configured	255	1000	Three partial CDRs are generated with 255 secondary RAT records each.
					Remaining reported Secondary RAT records become a part of CDR in the next trigger.
	S-GW	Not configured	32	1000	No partial CDR is generated.
					32 Secondary RAT records become part of the CDR in the next trigger.

Use the following configuration to control the maximum number of entries.

```
configure
  context context_name
    gtpp group group_name
      gtpp limit-secondary-rat-usage usage_limit
      default gtpp limit-secondary-rat-usage
```

```
no gtpp limit-secondary-rat-usage
end
```

NOTES:

- **gtpp limit-secondary-rat-usage** *usage_limit*: Enter a maximum number of secondary RAT reports. *usage_limit* must be an integer in the range of 1-100. The recommended value for S-GW CDR is 32.

  For example, if the limit is set to 10, then the CDR is generated once the configured value is reached.

- **default gtpp limit-secondary-rat-usage**: Specifies a default value of 32.

- **no gtpp limit-secondary-rat-usage**: Disables the CDR generation with limited number of secondary RAT usage information.

Suppressing Zero-Volume Secondary RAT Usage Report

Use the following configuration to suppress zero-volume Secondary RAT Usage report.

```
configure
  context context_name
    gtpp group group_name
      gtpp suppress-secondary-rat-usage zero-volume
      default gtpp suppress-secondary-rat-usage zero-volume
      no gtpp suppress-secondary-rat-usage zero-volume
      end
```

NOTES:

- **gtpp suppress-secondary-rat-usage zero-volume**: Suppresses either Secondary RAT records or zero volume Secondary RAT records.

- **default gtpp suppress-secondary-rat-usage zero-volume**: Does not suppress the zero volume secondary RAT usage records.

- **no gtpp suppress-secondary-rat-usage zero-volume**: Does not suppress the zero volume Secondary RAT usage records.

Monitoring and Troubleshooting

This section provides information on how to monitor and troubleshoot using show commands available in support of this feature.

Show Commands and Outputs

This section provides information regarding show commands and their outputs for this feature.

show config

The output of this CLI command displays the following parameters.

Field	Description
gtpp attribute secondary-rat-usage	Specify this option to include the Secondary RAT reports field in the CDR.
gtpp suppress-secondary-rat-usage zero-volume	Enables the exclusion of the zero volume Secondary RAT reports in the CDR.
gtpp limit-secondary-rat-usage	Enables limiting the number of Secondary RAT Usage reports in CDR with the configured value.

show config verbose

The output of this CLI command displays the following parameters.

Field	Description
gtpp attribute secondary-rat-usage	Displays the Secondary RAT usage records.
gtpp suppress-secondary-rat-usage zero-volume	Displays only Secondary RAT records that is having non-zero volumes from P-GW and S-GW.
gtpp limit-secondary-rat-usage	If total received Secondary RAT records are multiples of 10, displays multiple CDR generated by P-GW and S-GW. The reported cause value will be the maximum change condition.
no gtpp limit-secondary-rat-usage	Displays Secondary RAT records for unconfigured cause.

show gtpp group

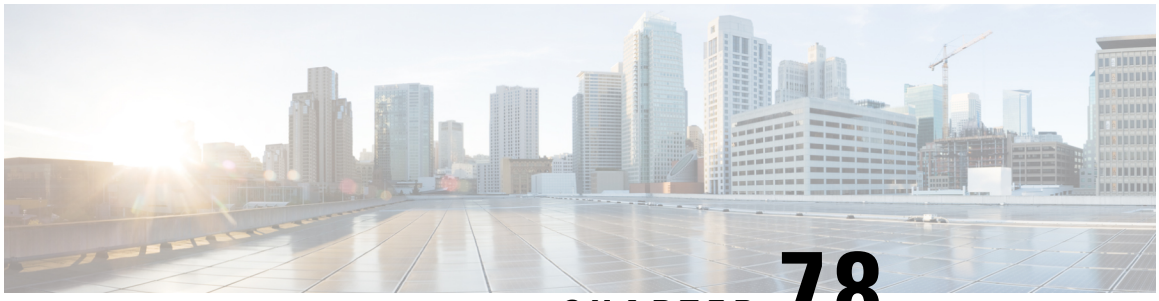
The output of this CLI command displays the following parameters.

Field	Description
Secondary RAT records present	Specifies whether the Secondary RAT record is present or not. The available options are: • no • yes
Limit-secondary-rat-usage	Specifies a limit for Secondary RAT usage report.

show gtpp statistics group

The output of this CLI command displays the following parameter.

Field	Description
Total PGW-CDR exceed size limit	Displays the total number of CDRs that exceeded size limit in P-GW.



CHAPTER 78

Self-overload Detection and Admission Control of Sx at UP

- Revision History, on page 707
- Feature Description, on page 707
- Configuring Overload Control at User Plane, on page 708
- Monitoring and Troubleshooting, on page 710

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Overload detection and control at User Plane (UP) is implemented using the eMPS functionality. During an overload scenario at Sx, the Session Establishment and Modification requests that are received at Sx (UP) are rejected for all non-eMPS subscribers.

Currently, overload control is supported for Sx Control Plane (CP). To support eMPS at UP, the CP adds the eMPS value to Message Priority IE in the PFCP header and sends the message over to UP.

The UP, on receipt of the Sx Session Establishment/Modification request, performs an overload check. If the detected system load is normal, the session establishment/modification is allowed, and the session is marked as a priority session based on the MP flag set in the PFCP header.

If the detected system load is overloaded, the Sx Session Establishment/Modification is rejected for all eMPS subscribers.

The system load level is determined by the following factors:

- System Utilization (CPU, Memory, and Licenses)

- Session Manager Utilization (CPU and Memory)
- VPP-CPU Utilization

Limitations

The following are the known limitations of the feature:

- Data throttling is not supported.
- Alarms are not supported.
- Bulk statistics are not supported.
- No support for handling APN-based emergency calls in a Pure-S scenario. Other emergency calls such as – IMSI-based and IMEI-valid based are handled.
- Only self-overload protection is supported in this release.
- User Plane ICSR not supported in this release.
- Impact on existing calls: If **userplane-overload-control-profile** is configured and associated to user plane service. Also, if the system moves to overload condition and the user plane service rejects SX Session Establishment and SX session modification messages, this leads to call cleanup/drop of relevant calls triggering SX session modification messages. This behavior continues until the system returns to the normal load condition.

Configuring Overload Control at User Plane

eMPS Profile Creation and Association to S-GW and P-GW Services of Control Plane

Important This configuration must be done before configuring an overload control profile at UP.

```
configure
  emps-profile profile_name
    earp earp_value
    end
configure
  context context_name
    sgw-service service_name
      associate emps-profile profile_name
      exit
    pgw-service service_name
      associate emps-profile profile_name
      end
```

Configuring the Overload Control Profile at UP

Use the following commands to configure overload control profile.

```
configure
  userplane-overload-control-profile profile_name
  end
```

Configuring Overload Threshold Parameters

Use the following commands to configure overload threshold parameters.

```
configure
  userplane-overload-control-profile profile_name
    overload-threshold system lower-limit limit_value upper-limit limit_value
 sessmgr lower-limit limit_value upper-limit limit_value vpp-cpu lower-limit
 limit_value upper-limit limit_value
     end
```

NOTES:

- **overload-threshold**: Configures Overload thresholds limits for system, sessmgr and vpp-cpu.
- **system**: Configures overload system threshold after which node moves to self-protection mode.
- **vpp-cpu**: Configures the overload vpp-cpu threshold after which node moves to self-protection mode.
- **sessmgr**: Configures the overload threshold for session manager after which node moves to self-protection mode.
- **upper-limit** *limit_value*: Configures overload vpp-cpu threshold L2 after which node moves to self-protection mode. Default limit value is 60%.
- **lower-limit** *limit_value*: Configures overload vpp-cpu threshold L1 after which node moves to self-protection mode. Default limit value is 50%.

Configuring System Weightage Parameters

Use the following commands to configure session manager weightage parameters.

```
configure
  userplane-overload-control-profile profile_name
     system-weightage system-cpu-utilization utilization_value
system-memory-utilization utilization_value license-session-utilization
utilization_value
     end
```

NOTES:

- **system-weightage**: Configures system weightage for various overload control parameters. Total weightage of all the parameters should be 100. The default values are 40% weightage to system-cpu-utilization, 30% weightage to system-memory-utilization and 30% weightage to license-session-utilization.
- **system-cpu-utilization**: Configures system CPU utilization weightage in percentage. Default weightage in overload factor is 40%.

- **system-memory-utilization**: Configures system memory utilization weightage in percentage. Default weightage in overload factor is 30%.
- **license-session-utilization**: Configures license session utilization weightage for User Plane service in percentage. Default weightage in overload factor is 30%.

Configuring Session Manager Weightage Parameters

Use the following commands to configure session manager weightage parameters.

```
configure
  userplane-overload-control-profile profile_name
    sessmgr-weightage sessmgr-cpu-utilization utilization_value
sessmgr-memory-utilization utilization_value
    end
```

NOTES:

- **sessmgr-weightage**: Configures sessmgr weightage for various overload control parameters. Total weightage of all the parameters should be 100. The default values are 35% weightage to sessmgr-cpu-utilization and 65% weightage to sessmgr-memory-utilization.
- **sessmgr-cpu-utilization**: Configures session manager CPU utilization weightage in percentage. Default weightage in overload factor is 35%.
- **sessmgr-memory-utilization**: Configures session manager memory utilization weightage in percentage. Default weightage in overload factor is 65%.

Associating an Overload Control Profile with a User Plane Service

Use the following commands to associate the Overload Control profile to a use plane service.

```
configure
  context context_name
    user-plane-service service_name
    [ no ] associate userplane-overload-control-profile profile_name
```

NOTES:

- **associate**: This command associates the user plane overload control profile with a user plane service.

Monitoring and Troubleshooting

Show Commands Input and/or Outputs

This section provides information regarding show commands and their outputs in support of the feature.

show user-plane-service name *name*

The following fields are displayed in support of this feature:

- Service name
  - Service-Id
  - Context
  - Status
  - PGW Ingress GTPU Service
  - SGW Ingress GTPU Service
  - SGW Egress GTPU Service
  - Control Plane Tunnel GTPU Service
  - Sx Service
  - Control Plane Group
  - Userplane Overload Control Profile
  - Fast-Path service

show user-plane-service statistics name *user_plane_service_name*

The following fields are displayed in support of this feature:

- Overload Control Information
  - Current Overload Factor System: Average of all user plane service values
  - Current Overload Factor SessMgr
  - Current Overload Factor VPP-CPU
  - No of times Overload Threshold Reached
  - No of Session Eshtablishment Req rejected during overload
  - No of Session Modif Req rejected during overload
  - No of eMPS Session Eshtablishment Req allowed during overload
  - No of eMPS Session Modif Req allowed during overload

show userplane-overload-control-profile name *name*

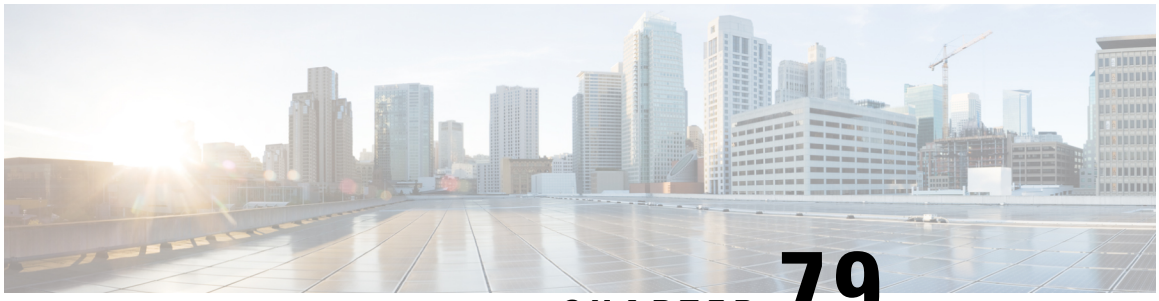
The following fields are displayed in support of this feature:

- User Plane Overload Control Profiles
- User Plane Overload Control Profile Name
- System Weightage and Thresholds:
  - CPU Utilization Weightage
  - Memory Utilization Weightage

- License Session Utilization Weightage
- System Threshold Lower Limit
- System Threshold Upper Limit

- Sessmgr Weightage and Thresholds:
  - CPU Utilization Weightage
  - Memory Utilization Weightage
  - Sessmgr Threshold Lower Limit
  - Sessmgr Threshold Upper Limit

- VPP Weightage and Thresholds:
  - VPP Utilization Weightage
  - vpp-cpu Threshold Lower Limit
  - vpp-cpu Threshold Upper Limit



CHAPTER 79

Smart Licensing

- Revision History, on page 713
- Overview, on page 713
- Configuring Smart Licensing, on page 718
- Monitoring and Troubleshooting Smart Licensing, on page 719

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Overview

Ultra Packet Core CUPS supports Smart Licensing. Smart Licensing is a cloud-based approach to licensing that simplifies the purchase, deployment, and management of Cisco software assets. Entitlements are purchased through your Cisco account via Cisco Commerce Workspace (CCW) and immediately deposited into your Virtual Account for usage. This eliminates the need to install license files on every device. Products that are smart-enabled, communicate directly to Cisco to report consumption. A single location is available to customers to manage Cisco software licenses—the Cisco Smart Software Manager (CSSM). License ownership and consumption are readily available to help make better purchase decision based on consumption or business need.

See https://www.cisco.com/c/en/us/buy/smart-accounts/software-licensing.html for more information about Cisco Smart Licensing.

Comparison Between Legacy Licensing and Smart Licensing

Cisco employs two types of license models - Legacy Licensing and Smart Software Licensing. **Legacy Licensing** consists of software activation by installing Product Activation Keys (PAK) on to the Cisco product. A Product Activation Key is a purchasable item, ordered in the same manner as other Cisco equipment and

used to obtain license files for feature set on Cisco Products. **Smart Software Licensing** is a cloud-based licensing of the end-to-end platform leveraging few tools that authorize and deliver license reporting. Smart Software Licensing functionality incorporated into StarOS completes the product registration, authorization resulting in reporting services available to the end customer.

Evaluation Period

A 90-day evaluation period is granted for all licenses in use. During this period, feature licenses can be used without limitation, and up to one counting license each can be used. The evaluation period ends when the system registers successfully with the CSSM or Cisco.com. Licensed functionality is blocked when this 90-day period expires.

CUPS performs license enforcement for on/off feature licenses. Each on/off feature license is tied to service licenses, which potentially use those on/off features. When an Out of Compliance (OOC) is detected for an on/off license, new calls for the corresponding services will be dropped, subject to the following conditions:

- Each on/off feature license is given a 90-day grace (evaluation) period. During this period, the system generates SNMP traps to inform of the unavailability of valid licenses. To resolve the OOC, corrective action is needed such as purchasing and registering licenses for this feature, or disabling the feature.
- If the feature is still OOC after the 90-day grace period, CUPS enforces the OOC state based on a predefined policy for each license. If enforcement is required, new calls for the services corresponding to the on/off licenses are dropped.

The following CLI commands can be used to display details about the enforcement of Smart Licenses in use:

```
show license enforcement policy
show license enforcement status [ allowed | blocked ] [ feature | service
 ]
```

Cisco Smart Software Manager

Cisco Smart Software Manager (CSSM) enables the management of software licenses and Smart Account from a single portal. The interface allows you to activate your product, manage entitlements, and renew and upgrade software. A functioning Smart Account is required to complete the registration process. To access the Cisco Smart Software Manager, see https://software.cisco.com.

Smart Accounts/Virtual Accounts

A Smart Account provides a single location for all Smart-enabled products and entitlements. It helps speed procurement, deployment, and maintenance of Cisco Software. When creating a Smart Account, you must have the authority to represent the requesting organization. After submitting, the request goes through a brief approval process.

A Virtual Account exists as a sub-account withing the Smart Account. Virtual Accounts are a customer-defined structure based on organizational layout, business function, geography or any defined hierarchy. They are created and maintained by the Smart Account administrator.

See https://software.cisco.com to learn about, set up, or manage Smart Accounts.

Smart Licensing Mode

The Smart Licensing Mode is categorized as follows:

- **Reporting Licenses (Parent Licenses)**: The Parent Licenses are reported to backend license server (CSSM) and accounted for usage of licenses. For each Parent Licenses, the entitlement tags are created and the same is used to identify the type service or feature.
- **Non-Reporting Licenses (Child Licenses)**: The Child Licenses are not reported to backend license server (CSSM) and these licenses are enabled by default with the Parent Licenses. For Child Licenses, the entitlement tags are not created.

That is to say, Smart License enables all Parent and Child Licenses based on the Product Type that is configured. However, the reporting is done only for Parent Licenses.

The state of Smart Licensing Agent is persistent across reboot and crashes.

Request a Cisco Smart Account

A Cisco Smart Account is an account where all products enabled for Smart Licensing are deposited. A Cisco Smart Account allows you to manage and activate your licenses to devices, monitor license use, and track Cisco license purchases. Through transparent access, you have a real-time view into your Smart Licensing products. IT administrators can manage licenses and account users within your organization's Smart Account through the Smart Software Manager.

Step 1 In a browser window, enter the following URL:

```
https://software.cisco.com
```

Step 2 Log in using your credentials, and then click **Request a Smart Account** in the **Administration** area.

The **Smart Account Request** window is displayed.

Step 3 Under **Create Account**, select one of the following options:

- **Yes, I have authority to represent my company and want to create the Smart Account** – If you select this option, you agree to authorization to create and manage product and service entitlements, users, and roles on behalf of your organization.
- **No, the person specified below will create the account** – If you select this option, you must enter the email address of the person who will create the Smart Account.

Step 4 Under **Account Information**:

a) Click **Edit** beside **Account Domain Identifier**.
b) In the **Edit Account Identifier** dialog box, enter the domain, and click **OK**. By default, the domain is based on the email address of the person creating the account and must belong to the company that will own this account.
c) Enter the **Account Name** (typically, the company name).

Step 5 Click **Continue**.

The Smart Account request will be in pending status until it has been approved by the Account Domain Identifier. After approval, you will receive an email confirmation with instructions for completing the setup process.

Software Tags and Entitlement Tags

Tags for the following software and entitlements have been created to identify, report, and enforce licenses.

Software Tags

Software tags uniquely identify each licenseable software product or product suite on a device. The following software tags exist for CUPS.

Product Type / Description	Software Tag
CUPS_CP 4G CUPS - Control Plane	regid.2020-08.com.cisco.CUPS_CP, 1.0_7afd7a3c-38dd-4a04-aecc-26df25029649
CUPS_UP 4G CUPS - User Plane	regid.2020-08.com.cisco.CUPS_UP, 1.0_fd28551c-a541-4902-87af-bba2d6b33cf1

Reporting (Parent) Entitlement Tags for CUPS_CP

The following entitlement tags indentify licenses in use for each product type.

License Display Name/Description	Entitlement Tag	License Type	Reporting Slab	Tag Name
4G CUPS CP 1K 4G CUPS Control Plane 1K Sessions	regid.2020-08.com.cisco. L_CUPS_CP_SAE_1K, 1.0_a84e70b6-d3f9-41c9 -8449-4b7bb7426b30	Counting	1K	L_CUPS_CP _ SAE_1K

Reporting (Parent) Entitlement Tags for CUPS_UP

The following entitlement tags indentify licenses in use for each product type.

License Display Name/Description	Entitlement Tag	License Type	Reporting Slab	Tag Name
4G CUPS UP 1K 4G CUPS User Plane 1K Sessions	regid.2020-08.com.cisco. L_CUPS_UP_SAE_1K, 1.0_41005ab7-1ad0-46ac -905b-c3c5ed402981	Counting	1K	L_CUPS_UP _SAE_1K
4G CUPS UP Instances 4G CUPS User Plane Instances	regid.2020-08.com.cisco. F_CUPS_UP_INS, 1.0_897c46a0-04b5-4fdb -bedd-9d5fb75bdb76	On/Off	1/0	F_CUPS_UP_INS

Non-reporting (Child) License List

In this release, the following Child Licenses are enabled by default when the Parent Licenses are enabled.

License Description	License Type
PGW 1k Sessions	Counting
SGW 1k Sessions	Counting
GGSN 1k Sessions	Counting

License Description	License Type
Per Subscriber Stateful Firewall 1k Sessions	Counting
ENAT 1k Sessions	Counting
Enhanced Charging Bundle 1	Counting
Enhanced charging bundle 2	On/Off
Dynamic policy interface	On/Off
Enhanced LI service	On/Off
Lawful intercept	On/Off
Session recover	On/Off
Radius AAA server group	On/Off
IPv6	On/Off
Intelligent Traffic Control	On/Off
DIAMETER Closed-Loop Charging Interface	On/Off
Per-Subscriber Traffic Policing/Shaping	On/Off
Dynamic Radius extensions (CoA and PoD)	On/Off
Proxy MIP	On/Off
FA	On/Off
IPSec	On/Off
Inter-Chassis Session Recovery	On/Off
ICSR/SR Performance Improvements	On/Off
ICSR Enhanced Recovery for Data and Control Plane, 1K Sessions	On/Off
MPLS	On/Off
TACACS+	On/Off
NAT/PAT With DPI	On/Off
Rate Limiting Function (Throttling)	On/Off
Overcharging Protection for EPC-GW	On/Off
Overcharging Protection Upgrade for EPC-GW	On/Off
ADC Trigger Over Gx, 1K Sessions	On/Off
Gx Based Virtual APN Selection, 1K Sessions	On/Off
EPC-GW Support for Wi-Fi Integration, 1K Sessions	On/Off
EPC-GW Non-Standard QCI Support, 1K Sessions	On/Off
Local Policy Decision Engine	On/Off
Header Enrichment	On/Off

License Description	License Type
HTTP Header Encryption	On/Off
HTTP Header Enrichment and Encryption	On/Off
Broadcast & Multicast Services	On/Off
Integrated Content Filtering Provisioned Service	On/Off
Application Detection and Control 1k Sessions	Counting
5G NSA Feature Set 100K Sess VPCSW Active 1k Sessions	Counting
5G NSA Enablement Fee, Network Wide	On/Off
Multimedia Priority Service Feature Set,1K Sessions	On/Off
EPC Gw VoLTE enhancements	On/Off
DNS Snooping	On/Off

Configuring Smart Licensing

Before you begin, ensure you have:

- Created a Smart Licensing account on https://software.cisco.com.
- Registered your products on https://software.cisco.com using the Product Instance Registration tokens created as part of Smart Account/Virtual Account.
- Enabled a communication path between the StarOS system to the CSSM server or Cisco.com.

Enable Smart Licensing

By default, Smart Licensing is disabled in CUPS. To enable Smart Licensing, enter the following Config mode commands:

```
configure
   license smart product { cups-cp | cups-up }
   license smart enable
   end
```

NOTE: Before enabling Smart Licensing, Product Type must be configured to enable default licenses that are based on product type.

Enter the following command to verify the configuration:

```
show configuration | grep license
```

Register the Device with Cisco

Using the Product Instance Registration token ID provided when you registered the products on https://software.cisco.com, register the system using the following Exec mode command:

```
license smart register idtoken token
```

The system now automatically reports entitlement usage count to the CSSM server and receives a compliance status. This also removes the system from "Evaluation Mode".

To show the compliance status, enter any of the following Exec mode commands:

```
show license status
show license summary
show license statistics
```

The registration for the system is renewed automatically every 180 days. If needed, use the following Exec mode command to renew the registration information manually:

```
license smart renew id
```

The license authorization for the system is renewed automatically every 30 days. If needed, use the following Exec mode command to renew the license authorization manually:

```
license smart renew auth
```

To unregister a device, enter the following Exec mode command:

```
license smart deregister
```

Changing Smart Transport URL

Smart Agent uses Smart Transport to communicate to Cisco CSSM server. Smart Transport uses the configured URL to identify destination URL where CSSM is reachable. This will not initiate any communication with Cisco. If needed, enter the following Configuration mode commands:

```
configure
   license smart transport smart
   license smart url https_link
```

Handling Out of Compliance

If there are not enough licenses in the virtual account for a given SKU, CSSM sends an Out Of Compliance (OOC) message to the device. The system stops allowing additional sessions until the OOC state is cleared. The OOC state is cleared when the device receives an authorized response.

Monitoring and Troubleshooting Smart Licensing

Enter the following Exec mode command to verify the Smart Licensing configuration:

```
show configuration | grep license
```

The following Exec mode commands display information about Smart Licensing:

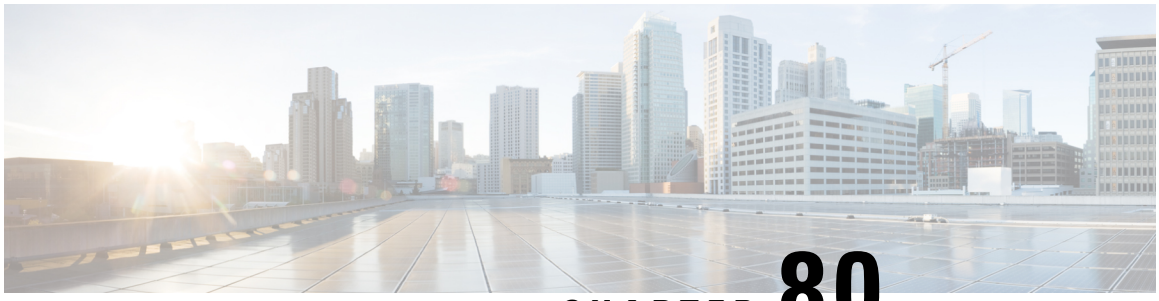
```
show license { all | enforcement | smart-tags | statistics | status |
summary | tech-support | udi | usage }
```

NOTES:

- **all** - Shows a superset of information that includes show status, show usage, show UDI, as well as the Smart Licensing agent version.
- **enforcement { policy | status [allowed | blocked] [feature | service] }** - Shows the enforcement policy applied or current enforcement status of Smart Licenses. Status information can be filtered to

show only the licenses which are currently allowed or blocked, or by type (feature license or service license).

- **smart-tags [feature | service]** - Shows the features and services that are currently supported and the corresponding Smart Entitlement Tag.
- **statistics [verbose]** - Shows individual feature license status.
- **status** - Shows overall Smart Licensing status information.
- **summary** - Shows summary of Smart Licensing status.
- **tech-support** - Shows information useful for debugging issues with Smart Licensing.
- **udi** - Shows details for all Unique Device Identifiers (UDI).
- **usage** - Shows the usage information for all entitlements that are currently in use.



CHAPTER 80

Software Management Operations

- Revision History, on page 721
- Overview, on page 721
- Upgrading or Downgrading of CP and UP, on page 723

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
Support is extended for N-4 backward compatibility of software releases.	21.26
Support is extended for N-3 backward compatibility of software releases.	21.25
Support is extended for N-2 backward compatibility of software releases.	21.24.1
First introduced.	Pre 21.24

Overview

CUPS supports backward compatibility of software releases on Control Plane (CP) and User Plane (UP). The feature allows seamless upgrade/downgrade of the software from/to one previous release (N-1)/two previous releases (N-2)/three previous releases (N-3)/four previous releases (N-4). The functionality includes support for the following:

- N-1/N-2/N-3 /N-4 compatibility of software releases on two CPs in ICSR mode—allows seamless upgrade of CPs from one version to another in CP 1:1 redundancy scenario.
- N-1/N-2/N-3 /N-4 compatibility of software releases on two UPs in ICSR mode—allows seamless upgrade of UPs from one version to another in UP 1:1 redundancy scenario.
- N-1/N-2/N-3/N-4 compatibility of software releases between CP and UP—allows seamless upgrade of the associated CP or UP from one version to another.

- N-1/N-2/N-3/N-4 compatibility of software releases between CP and UP with multi-Sx—allows seamless upgrade of the associated CP or UP from one version to another in multi-Sx scenario.

Important Contact your Cisco Account representative for procedural assistance prior to upgrading or downgrading your software versions.

Version Exchange between CP and UP

Version/release information is exchanged when CP and UP pairs. The release information exchange also occurs when the CP pairs with a Standby CP or UP pairs with a Standby UP (in 1:1 redundancy scenario) through the heart beat message exchanged between Active and Standby.

When incompatible releases are paired, an Alarm (SNMP Trap) is raised. For details, see *SNMP Trap* section.

To indicate the peer version during the exchange of release information, the following new IE is included in the association request and heartbeat request messages.

Information Elements	P	Condition / Comment								IE Length	IE ID
Peer Version	O	Used to specify the peer GR/PFCP version and StarOS version								4 bytes	245
		Bits									
	Octets	8	7	6	5	4	3	2	1		
	1 to 2	Peer Version IE Type = 245 (decimal)									
	3 to 4	Length = n bytes									
	5 to 8	Peer GR/PFCP Version									
	9 to 12	StarOS GR Version									
	13 to 13	StarOS Version String Length									
	Variable Length	StarOS Version String Value									

SNMP Traps

The following SNMP traps are raised when pairing is done with an incompatible release.

SNMP Trap	Description
SRPPeerUnsupportedVersion	The Active/Standby CP/UP in higher version raises the SNMP trap when the peer is in a version lower than N-4.
SRPPeerUnsupportedVersionClear	The Active/Standby CP/UP in higher version raises the SNMP trap to clear the SRPPeerUnsupportedVersion.
SxPeerUnsupportedVersion	The CP/UP in higher version raises the SNMP trap when the peer is in a version lower than N-4.

SNMP Trap	Description
SxPeerUnsupportedVersionClear	The CP/UP in higher version raises an SNMP trap to clear the SxPeerUnsupportedVersion.

Limitations

The following are the known limitations of the feature:

- When the peer version is determined to be lower than the supported N-4 version, the association/pairing is allowed. However, functional aspect of the same isn't guaranteed.

Caution Don't attempt to upgrade from incompatible versions. Contact your Cisco Account representative for the upgrade path and steps.

SNMP traps are raised by the node on the latest version with respect to the StarOS version. For details, see the *SNMP Trap* section of this chapter.

- From release 21.24.1, RCM is checkpoint agnostic to enable support for future UP releases. Currently RCM does not support N-4 compatibility and only supports N-1 compatibility.

Upgrading or Downgrading of CP and UP

The following Maintenance Operating Procedure (MOP) outlines the steps necessary to Upgrade or Downgrade a Control Plane and User plane from previous release (N-1)/(N-2)/(N-3)/(N-4) to or from the latest N release.

Important Contact your Cisco Account representative for procedural assistance prior to upgrading or downgrading your software versions.

The following are the Upgrade options:

- **Only CP Upgrade**: When requirement is to upgrade only the CP, and the UP must remain intact.
- **Only UP Upgrade**: When requirement is to upgrade only the UP, and the CP must remain intact.
- **Both CP and UP Upgrade**: When requirement is to upgrade both CP and UP. In this case, upgrade the CP first and then the UP or the other way around.

Health Checks

Perform the following health checks after every operation of upgrade, downgrade, or reload of chassis.

1. Check the Service Redundancy Protocol (SRP) information on the Active chassis to avoid issues during an SRP switchover and decide if proactive analysis must be done before the SRP switchover. Use the following CLI commands:
   - **srp validate-configuration srp validate-switchover**

- **show srp info**

The following is a sample output.

```
Peer Configuration Validation: Complete
Last Peer Configuration Error: None
Last Peer Configuration Event: Wed Mar 18 15:34:02 2019  (1602 seconds ago)
Last Validate Switchover Status: None
Connection State: Connected
```

Check the following parameters:

- **Peer Configuration Validation: Complete**—If it shows "In Progress", you must wait and then execute the **show srp info** again after 15 seconds (approximately).
- **Last Peer Configuration Error: None**—If you see "Peer Checksum Validation Failure", then it indicates that there are configuration differences between Active and Standby chassis that must be fixed.
- **Last Validate Switchover Status: None**—The output must show as "None". And, output should be *Remote Chassis - Ready for Switchover (XX seconds ago)* when the **srp validate-configuration** and **srp validate-switchover** CLI commands are triggered.
- **Connection State: Connected**—The output must show as "Connected".

2. Check subscriber count on both Active and Standby chassis.

   After sessions are up, execute **show subscribers summary | grep Total** CLI command in the Active chassis. The following is a sample output.

   ```
   show subscribers summary | grep Total
      Total Subscribers:  100
   ```

   On Standby chassis, execute **show srp checkpoint statistics | grep allocated** CLI command. The following is a sample output.

   ```
   show srp checkpoint statistics | grep allocated
      Current pre-allocated calls:  100
   ```

3. Check the status of the license by executing **show license information** CLI command. It should be in "Good (Redundant)" and not in "Expired" state.
4. Check the Session Recovery Status by executing the **show session recovery status verbose** CLI command. The following is a sample output.

   ```
   Session Recovery Status:
     Overall Status        : Ready For Recovery
     Last Status Update    : 7 seconds ago

                 ----sessmgr---  ----aaamgr----  demux
    cpu state    active standby  active standby  active  status
   1/0  Active     8      1        8      1        17     Good
   ```

5. Verify all the SessMgrs are in Standby-Connected state on Standby chassis by executing **show srp checkpoint statistics | grep Sessmgrs** CLI command. The following is a sample output.

   ```
   Number of Sessmgrs:        1
   Sessmgrs in Active-Connected state:  0
   Sessmgrs in Standby-Connected state: 8
   Sessmgrs in Pending-Active state:    0
   ```

6. Verify the status of all the cards to see if they are in Active or Standby state. The following is a sample output.

```
show card table

Slot          Card Type              Oper State  SPOF  Attach
----------- ---------------------- ----------- ----- ------
1: VC        5-Port Virtual Card     Active        -
```

7. Execute **show task resources | grep -v good** CLI command and its output must only display the total number of SessMgrs and sessions.
8. Execute the **show crash list** CLI command to check if there were any new crashes.
9. Execute the **show service all** CLI command to verify that the state is displayed as "Started" and not "Initialized".

Build Upgrade

Backup Configuration

1. Back up the current configuration—save current configuration, that will be used in case of downgrade, which probably has all the features and configuration present until now.
2. Collect the **show support details** on both Active and Standby chassis before making any changes or Upgrade.
3. Perform Health Checks.

Upgrade Procedure

1. Perform chassis Health Checks on both the nodes.
2. On the secondary chassis (ICSR), which is in Standby state, change boot priority with N build.
3. Reload to latest 21.xx.xx build.
4. Do the new configuration change on Standby chassis (For example, any new CLI, license, or configuration changes.).
5. Do Health Check on the reloaded chassis. Check for any crashes or errors.

Perform Switchover

1. Before SRP switchover from Active to Standby on both chassis, check:
   a. On Active chassis: **show subscriber summary | grep Total**
   b. On Standby chassis: **show srp checkpoint statistics | grep allocated**

Note The count must be same for both.

   c. On Active and Standby chassis: **show sx peer**

For example:

```
|||||  Sx Service                                                  No of
|||||  ID                                                                  Restart
|||||  |                                                  Recovery              |
Current    Max        Peer
vvvvv  v     Group Name    Node ID         Peer ID     Timestamp             v
Sessions   Sessions   State
----- ---- ----------------- ------------------------- ------------------- ------
--------- --------- --------
CAAXD  22    CPGROUP21   209.165.200.225   50331649    2021-03-17:02:33:55    0
    0          0      NONE

Total Peers:    1
```

Note Peer state must be Active and associated. Peer ID must match on both the chassis.

d. On Standby chassis: **show srp checkpoint statistics | grep Sessmgrs**

Note "Number of Sessmgrs" must be equal to the "Sessmgrs in Standby- Connected state".

e. On Active chassis:

1. **srp validate-configuration**: This CLI command is for initiating a configuration validation check from the ACTIVE chassis. If the validation doesn't have any error, output of this CLI command is blank.
2. **srp validate-switchover**: Validates both Active and Standby chassis are ready for a planned SRP switchover. If chassis is ready for switchover, then the output of this CLI command is blank.
3. **show srp info | grep "Last Validate Switchover Status"** : Output of this CLI command must be as follows.

   ```
   Last Validate Switchover Status: Remote Chassis - Ready for Switchover
   ```

4. **show srp info debug**: Active and Standby chassis must have the same output.

2. On Active chassis: **srp initiate-switchover**

   a. Perform chassis Health Checks on both the nodes. Also check Step 1a and Step 1c under the *Performing Switchover* section. There can be a difference of 5%.

   b. Perform call testing since new sessions are serviced on the new Active chassis.

   c. Upgrade the old Active as mentioned in Step 2 through Step 5 under the *Upgrade Procedure* section.

CP Upgrade

This section describes the procedure for CP-only upgrade.

1. Perform Health Check procedure on both CP nodes as mentioned in the Health Checks, on page 723 section.

2. Perform Upgrade on Standby CP as mentioned in the Build Upgrade, on page 725 section.

Note If the context names on CP and UP are different, then execute **debug pgw pfd-mgmt** CLI command on the upgraded CP before making it Active.

3. Perform Health Check on both chassis, and then do CP switchover to the upgraded chassis.
4. Verify that the new chassis is taking the new sessions, there are no new crashes, or session drop due to error scenarios. Do Health Check on both the CP and UP.
5. Upgrade the new Standby CP as mentioned in the Build Upgrade, on page 725 section.

UP Upgrade

This section describes the procedure for UP-only upgrade.

1. Perform Health Check procedure on both the UP nodes as mentioned in the Health Checks, on page 723 section.
2. Perform Upgrade on Standby UP as mentioned in the Build Upgrade, on page 725 section.
3. Do "sx-peer configuration" on the upgraded Standby chassis.
4. Perform Health Check on both the UP nodes, and then do UP switchover.
5. Upgrade the new Standby UP as mentioned in the Build Upgrade, on page 725 section.

CP and UP Upgrade

This section describes the procedure for upgrading the CP first and then upgrading the UP, or the other way round.

Upgrading CP First

1. Perform Health Check procedure on both CP and UP, as mentioned in the Health Checks, on page 723 section.
2. Perform Upgrade on Standby CP as mentioned in the Build Upgrade, on page 725 section.

Note If the context names on CP and UP are different, then execute **debug pgw pfd-mgmt** CLI command on the upgraded CP before making it Active.

3. Perform Upgrade on Standby UP as mentioned in the Build Upgrade, on page 725 section.
4. Upgrade both Standby CP and UP to N build.
5. Perform Health Check on both chassis, and then do CP switchover to the upgraded chassis.
6. Verify that the new chassis is taking the new sessions, there are no new crashes, or session drop due to error scenarios.

7. Perform Health Check on both the UP nodes, and then do UP switchover.
8. Perform Health Check on newly Active UP. Verify that there are no call drops, and data is flowing through the new chassis.
9. Upgrade new Standby CP and UP as mentioned in the Build Upgrade, on page 725 section.

Upgrading UP First

1. Perform Health Check and build transfer procedure on both CP and UP.
2. Perform Upgrade on Standby UP as mentioned in the Build Upgrade, on page 725 section.
3. Do "sx-peer configuration" on the upgraded Standby chassis.
4. Perform Health Check on both the UP nodes, and then do UP switchover.
5. Perform Upgrade on new Standby UP as mentioned in the Build Upgrade, on page 725 section.
6. Perform Upgrade on Standby CP as mentioned in the Build Upgrade, on page 725 section.
7. Perform Health Check on both the CP nodes, and then do CP switchover.

Note If the context names on CP and UP are different, then execute **debug pgw pfd-mgmt** CLI command on the CP.

8. Upgrade new Standby CP chassis. Perform Health Check.
9. Perform Health Check on both the Active and Standby UP.
10. If everything is working as expected, then do the configuration changes on the Standby CP first. Then do the similar changes on the Active CP and execute **push config-to-up all** CLI command. New changes are pushed to the new Active UP.

Downgrade Procedure

Downgrade – Both CP and UP

If there are new configurations and/or configuration changes needed on CP as part of Upgrade, then follow the steps to upgrade the UP first.

1. Do Health Check on both CP and UP.
2. Change boot priority to the N-1/N-2/N-3/N-4 build on the Standby UP. Reload Standby UP.
3. Do "sx-peer configuration" on downgraded Standby UP.
4. Do Health Check on both the UP nodes and then do UP switchover.
5. Perform Step 1 to Step 3 on new Standby UP.
6. Change boot priority to N-1/N-2/N-3/N-4 build on the Standby CP. Reload the Standby CP.

Note If the context names on CP and UP are different, then execute the **debug pgw pfd-mgmt..** CLI command on the CP.

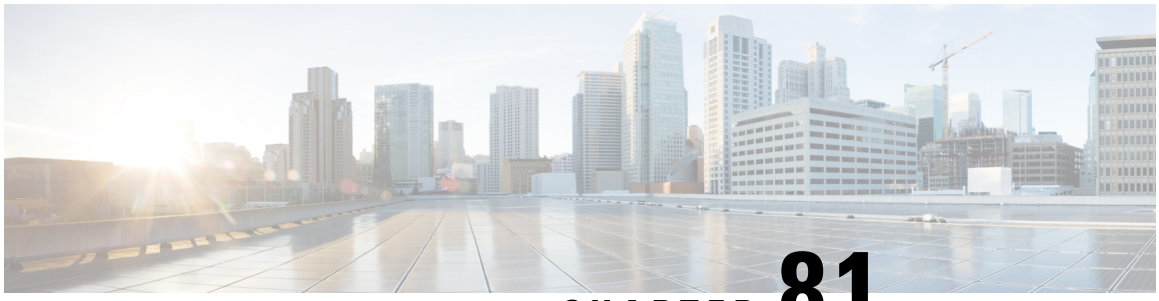
7. Load the configuration that has been saved in Step 1 mentioned in *Backup Configuration* section in Build Upgrade, on page 725.
8. Do Health Check on both CP nodes and then do the CP switchover.
9. Perform Step 6 and Step 7 to downgrade old Active.
10. On Active CP, execute **push config-to-up all** CLI command so that changes in the configuration are pushed to the UP.

Downgrade – CP Only

Perform Step 6 through Step 10 mentioned in the *Downgrade – Both CP and UP* section.

Downgrade – UP Only

Perform Step 1 through Step 5 mentioned in the *Downgrade – Both CP and UP* section.



CHAPTER 81

Standard QCI Support

- Revision History, on page 731
- Feature Description, on page 731

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced.	Pre 21.24

Feature Description

Important Standard QCI is not supported and qualified in the CUPS architecture.

The standardized QCI values—65, 66, 69, and 70 support Mission Critical and Push-to-Talk (MC/PTT) applications. The standard QCIs are based on 3GPP TS 23.203 Release 12.

This feature supports the following functionality:

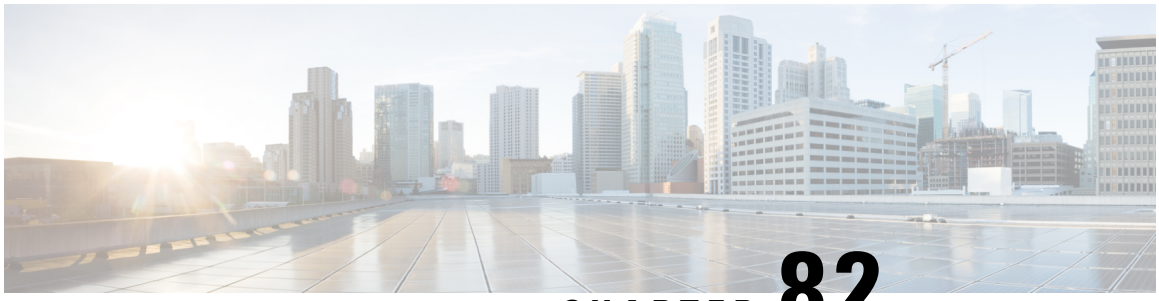
- Creates, deletes, and updates default and dedicated bearers.
- All applicable charging records include the standard QCI values.
- All features related to QCIs work with the standard QCI values.

Limitations

The following are the known limitations of this feature:

- Does not support S2a/S2b/GGSN.

- Does not support the overall eMPS functionality.



CHAPTER 82

Static and Predefined Rule Match Support for Shallow Packet Inspection

- Revision History, on page 733
- Feature Description, on page 733
- How It Works, on page 734
- Monitoring and Troubleshooting, on page 734

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

This feature adds support to check different data statistics related to the node or ongoing sessions in the CUPS deployment.

To support this functionality, a new keyword “real-time” has been added to the following CLI commands:

- show apn statistics real-time - Displays the aggregated data and control statistics across all APN’s from all user-planes connected to this control plane.
- show apn statistics real-time all - Displays independently per APN, the data and control statistics from all user-planes connected to this control plane.
- show apn statistics real-time name - Displays the data and control statistics by fetching data from all the user-planes for a given APN.

Important For this release, only the following eight counters are supported:

- Uplink Bytes
- Downlink Bytes
- Uplink Packets
- Downlink Packets
- Dropped Uplink Bytes
- Dropped Downlink Bytes
- Dropped Uplink Packets
- Dropped Downlink Packets

How It Works

The following points describe briefly how the SPI feature works:

- The static and predefined rule policies, which are available on the Control-Plane, are percolated to the User-plane based on the rulebase that is associated to the subscriber. This information is translated in the form of a PDR on the Control-Plane.

  Static and predefined rules on the Control-Plane that are translated to PDRs and sent to convert static rules into a rulebase PDR while the predefined rules would be translated as PDR IDs, individual PDR IDs and sent to the User-plane for activation. This is how a set of subscriber policies would be defined in the User-plane.

  The session establishment associates the predefined and static rule that is available to the subscriber. This handles the implementation of policies that are associated for a subscriber.

- The PDR match maps the data packet against the filters specified in the PDI field of the applicable PDRs. When all filter conditions are matched, the packet is matched to the PDR. Based on the FAR ID, the action to perform on the packet is known. Accordingly, the service chain is updated and executed.

- For static and predefined rules, a unique URR is generated based on the combination of QCI, service-ID, and rating-group is configured on the Control-Plane. This URR is passed on the User-plane and forwarding actions are implemented

  Based on this information, policing and charging actions are implemented for the packet, including updating URRs and applying QERs.

  For matched PDRs, the forwarding action is to either “allow” or “discard” a packet.

Monitoring and Troubleshooting

This section provides information regarding show commands and/or their outputs in support of this feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show subscribers user-plane-only full all

The output of this command has been enhanced to include the following new fields and values in support of this feature.

- Static & Predef Rule Match stats
  - Rule Name
  - Pkts-Down
  - Bytes-Down
  - Bytes-Up
  - Hits
  - Match-Bypassed
- Dynamic Rule Match stats
  - PDR Id
  - Pkts-Down
  - Bytes-Down
  - Pkts-Up
  - Bytes-Up
  - Hits
  - Match-Bypassed

show subscribers user-plane-only callid <callid> pdr full all

The output of this command has been enhanced to include the following field in support of this feature.

Rule Name

This field is displayed only for predefined rules.

show subscribers user-plane-only seid <seid> pdr full all

The output of this command has been enhanced to include the following field in support of this feature.

Rule Name

This field is displayed only for predefined rules.

show subscribers user-plane-only callid <callid> pdr id <id>

The output of this command has been enhanced to include the following field in support of this feature.

Rule Name

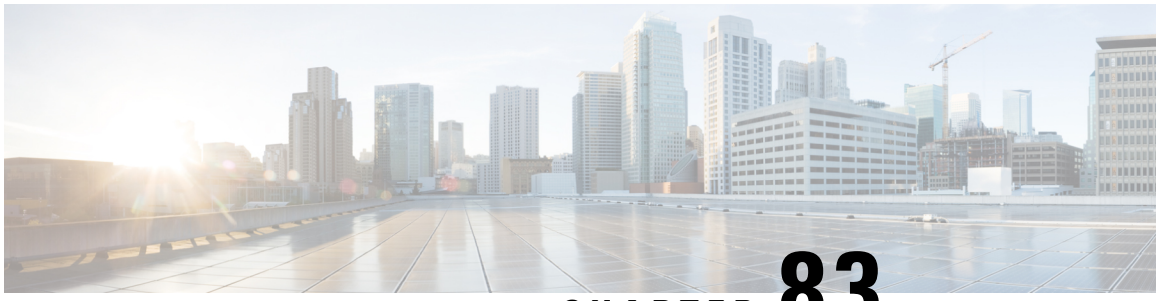
This field is displayed only for predefined rules.

show subscribers user-plane-only seid <seid> pdr id <id>

The output of this command has been enhanced to include the following field in support of this feature.

Rule Name

This field is displayed only for predefined rules.



CHAPTER 83

Static IP Assignment from RADIUS

- Feature Description, on page 737
- How it Works, on page 737

Feature Description

In this feature, static IP address for a subscriber is assigned from RADIUS server during the initial authentication procedure. This feature leverages the static IP address (UE-requested) functionality available in CUPS.

How it Works

After the RADIUS server assigns static IP address to the session, the User Plane selection of static session is fixed as per chunk allocation to User Plane from the User Plane group that is associated to an APN.

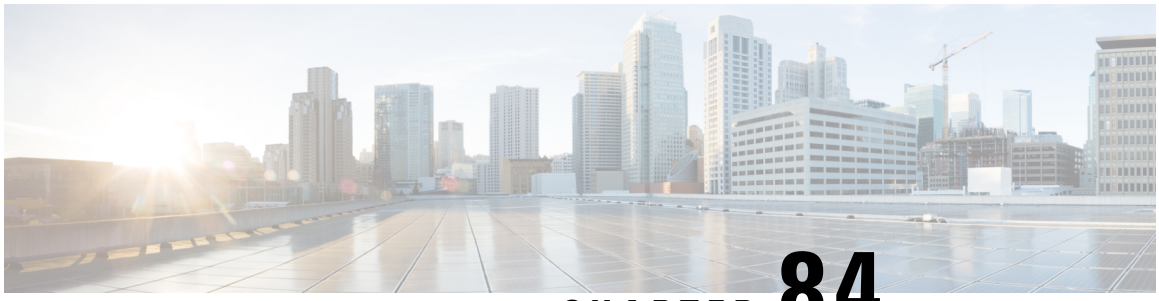
If same static IP address range is used across multiple APN, then it's recommended to use same User Plane group in those APN.

For more information on static IP pool management, refer the IP Pool Management chapter in the *Ultra Packet Core CUPS User Plane Administration Guide or Ultra Packet Core CUPS Control Plane Administration Guide*.

Limitations

The following are the known limitations of the feature:

- Static IP Address Pool assignment from RADIUS isn't supported as part of this feature.
- SAEGW-C doesn't support IPv4v6 PDN type call with static address received from RADIUS, even if one of the IP addresses (either IPv4 or IPv6, or both) is static address.
- SAEGW-C doesn't support allow-static type pool configuration.
- Multi-PDN call with static IP address allocation isn't supported.



CHAPTER 84

Suspend and Resume Notification for Pure-S Calls

- Revision History, on page 739
- Feature Description, on page 739
- How It Works, on page 739

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

Suspend and Resume Notifications for Pure-S calls are now supported in the CUPS architecture. The User Plane (UP) and Control Plane (CP) communicate through the Sx Establishment/Modification request when a Suspend/Resume notification is received.

Ongoing streams are maintained on the UP. When a Suspend/Resume notification is received, the CP changes the FAR action on UP through the Sx Modification request message. In response, the UP sets the appropriate FAR action.

On receiving a Modify Bearer request after a suspend notification, if an eNodeB TEID exists in the MBReq, the mode is set to Forward in the FAR. If the eNodeB TEID does not exist, then the mode is set to BUFFER.

How It Works

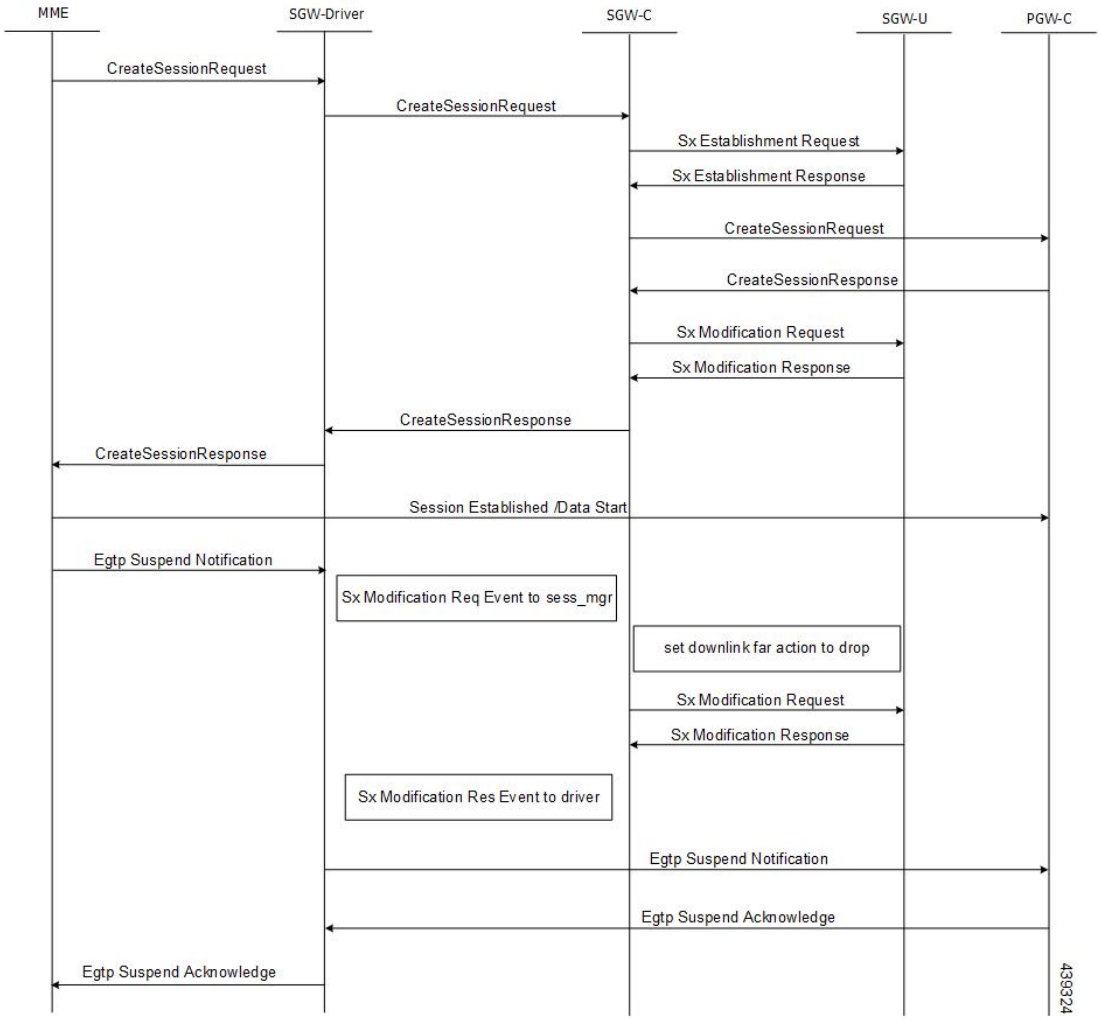
For a Suspend notification, downlink data is suspended by setting downlink FAR action to DROP. For a Resume notification, downlink data is buffered by setting downlink FAR action to BUFFER.

Call Flows

Suspend Notification

On receipt of a Suspend notification in Pure-S call, the SGW-C updates the Download FAR action by sending Sx Session Modification request to SGW-U with FAR action set as DROP.

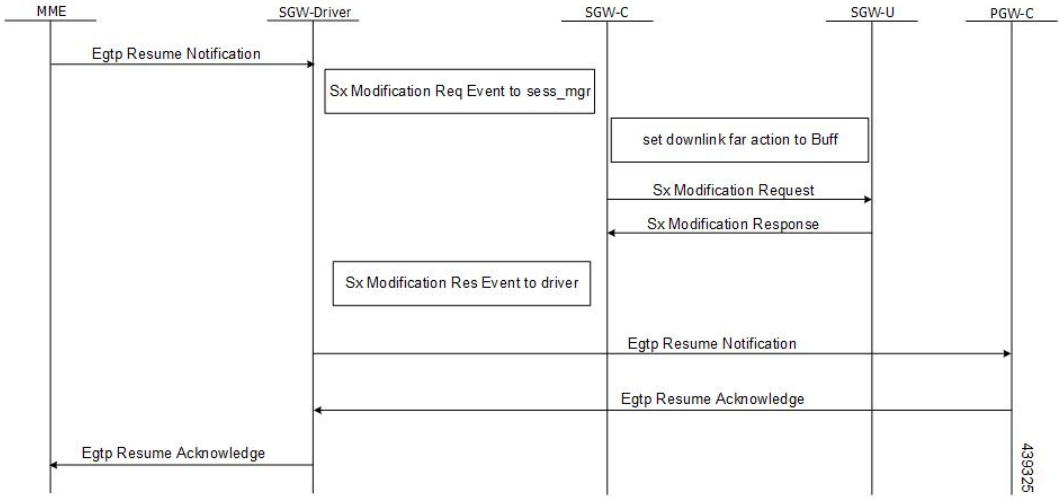
The following call flow, at a high level, illustrates the Suspend notification for Pure-S calls

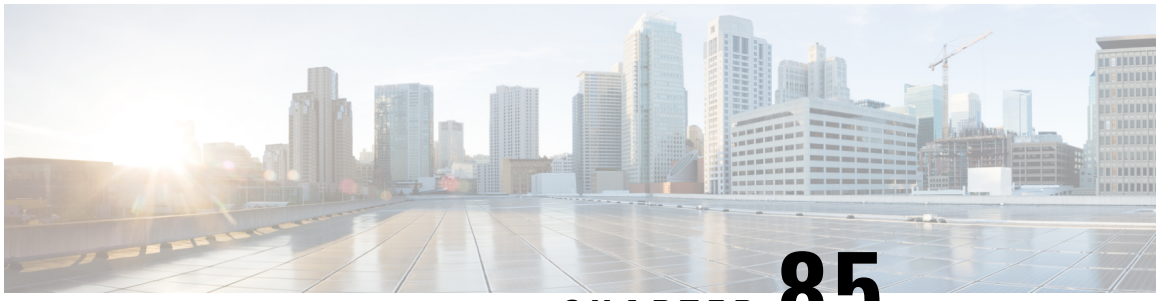


Resume Notification

On receipt of Resume notification in Pure-S call, the SGW-C updates the Download FAR action by sending Sx Session Modification request to SGW-U with FAR action set as BUFFER.

The following call flow, at a high level, illustrates the Resume notification for Pure-S calls.


MME
SGW-Driver
SGW-C
SGW-U
PGW-C
Egtp Resume Notification
Sx Modification Req Event to sess_mgr
set downlink far action to Buff
Sx Modification Request
Sx Modification Response
Sx Modification Res Event to driver
Egtp Resume Notification
Egtp Resume Acknowledge
Egtp Resume Acknowledge
439325



CHAPTER 85

TACACS+ Over IPSec

- Revision History, on page 743
- Feature Description, on page 743
- How it Works, on page 745
- Configuring TACACS+ over IPSec, on page 748
- Monitoring and Troubleshooting, on page 751

Revision History

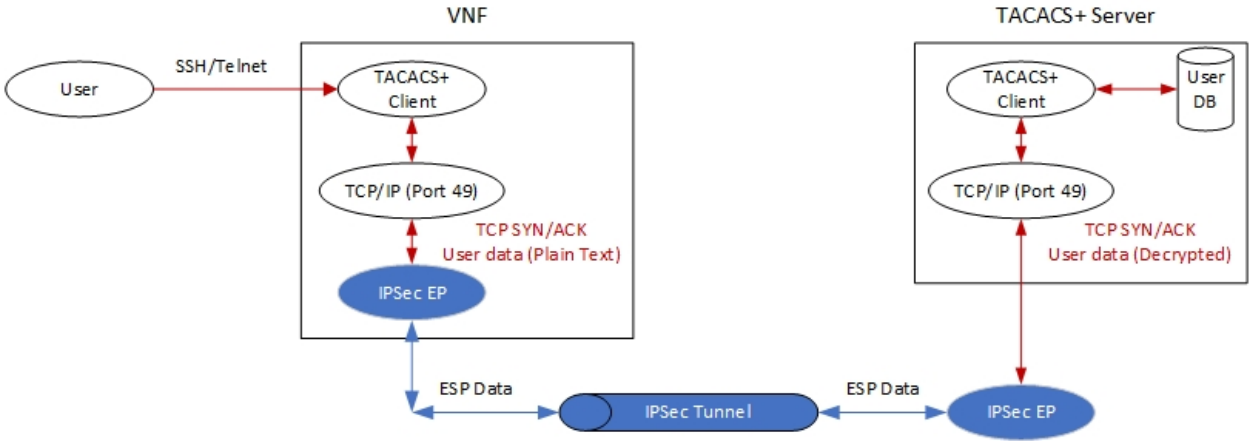
Revision Details	Release
First introduced	21.24

Feature Description

The Terminal Access Controller Access Control Server Plus (TACACS+) is a security protocol that is used for authenticating user access permissions on StarOS. To secure the authentication data that are sent over TACACS+ client and servers, CUPS VNFs support TACACS+ over IPSec for encrypting the authentication data.

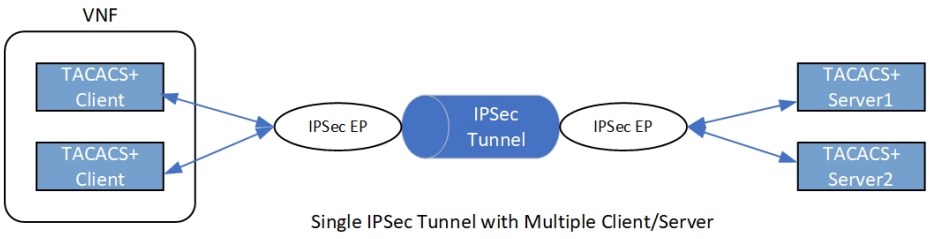
Architecture

The following diagram illustrates a secured TACACS+ architecture.

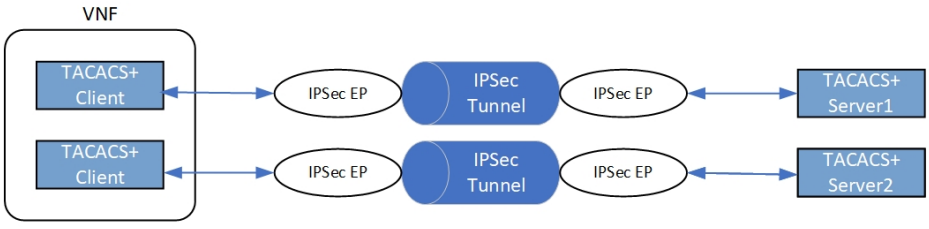


Deployment Architecture

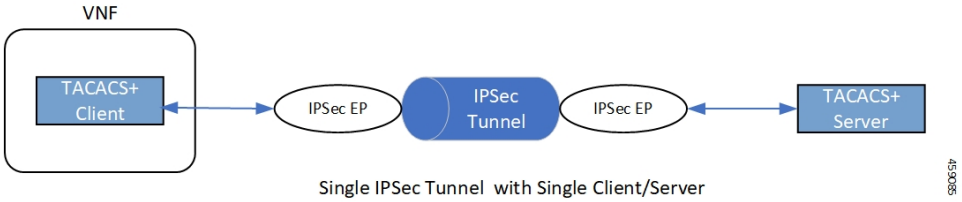
There are multiple ways you can use TACACS+ client/server in a secured way. You can either have single or multiple TACACS+ servers. A single VNF can host single or multiple clients. The TACACS+ over IPSec solution can handle multiple clients on a single VNF.



Single IPSec Tunnel with Multiple Client/Server



Multiple IPSec Tunnel with Multiple Client/Server



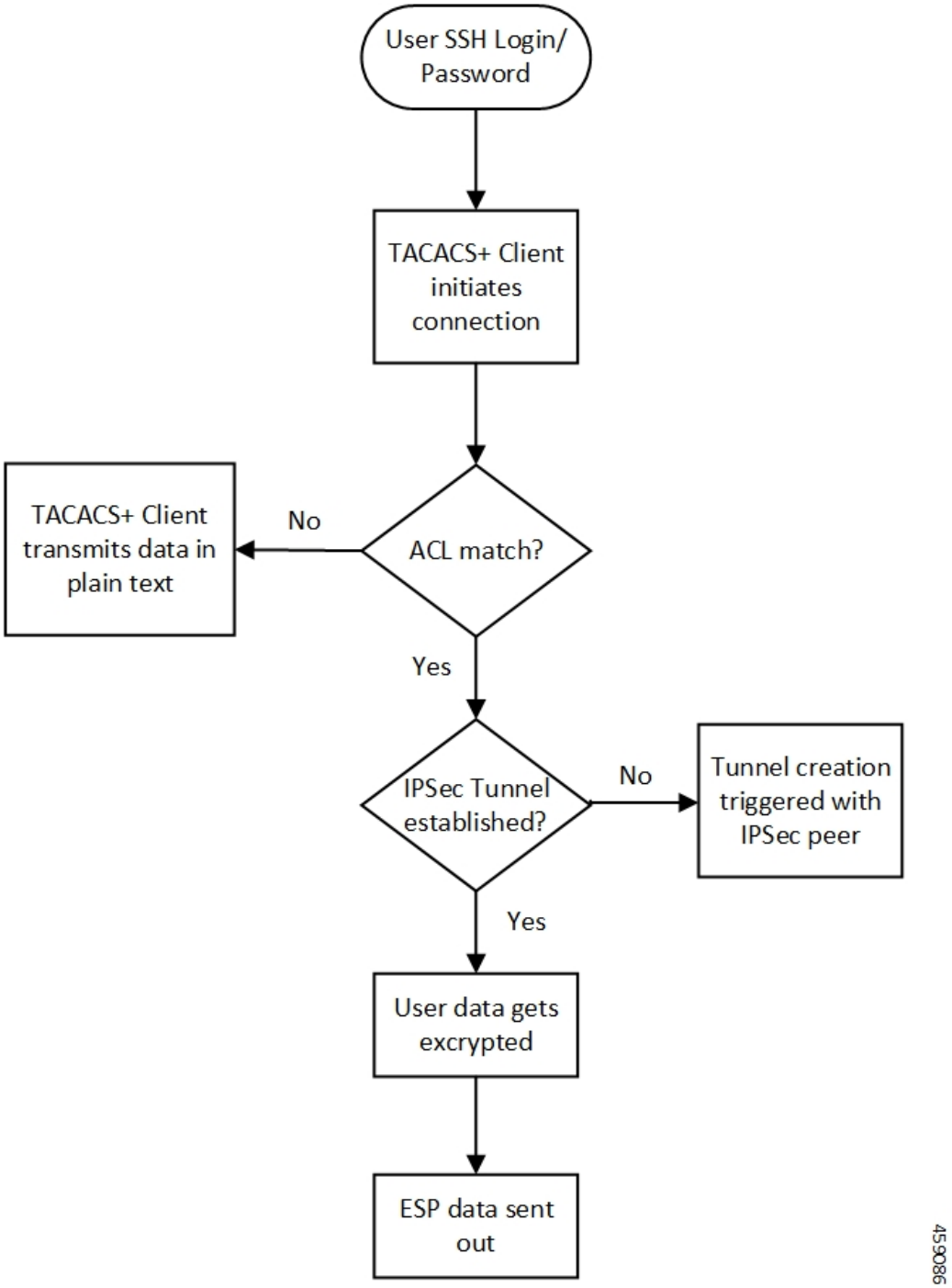
Single IPSec Tunnel with Single Client/Server

How it Works

Depending on the deployment requirement, multiple applications that must be secured has independent ACL rules configured as part of a single crypto-map or separate crypto-map. In both the cases, multiple TUN interfaces are created which are attached to each application requiring encryption.

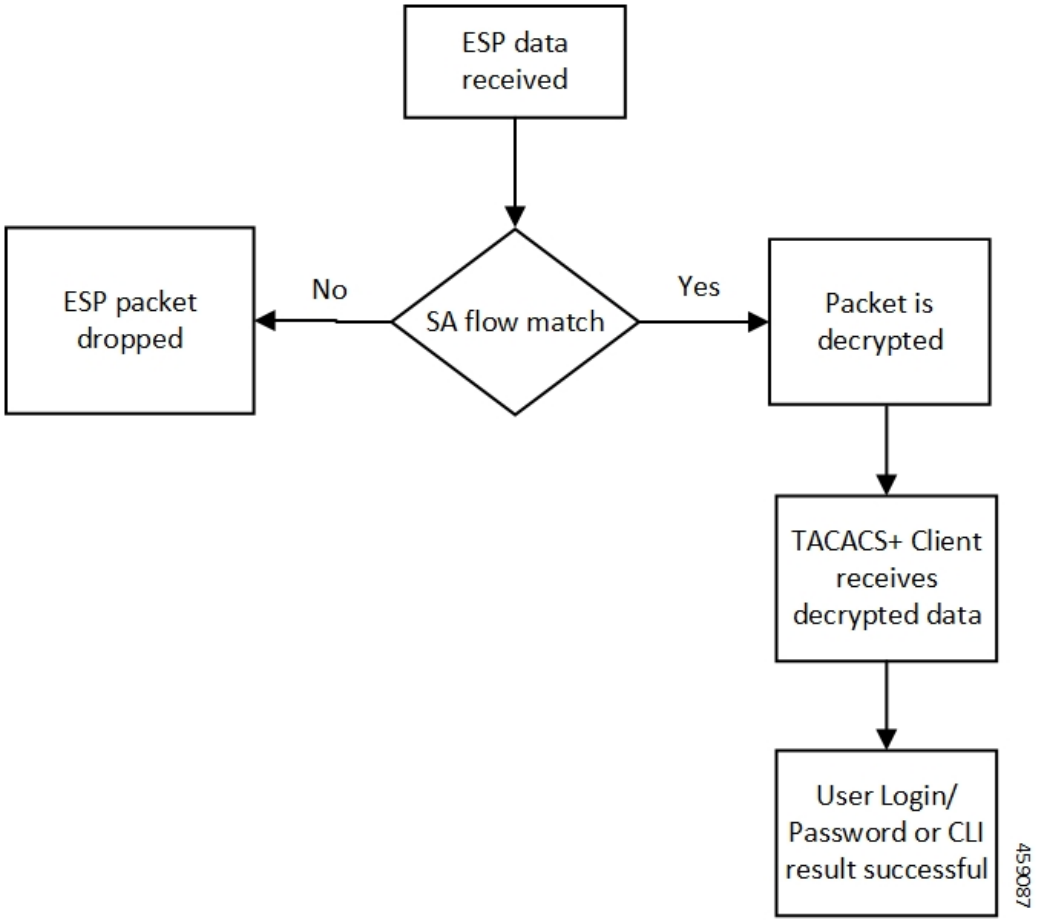
Encryption of TACACS+ Client Data

The following diagram illustrates the tunnel establishment and packet encryption.



Decryption of TACACS+ Server Data

The following diagram illustrates the packet decryption.



The following steps describe the packet flow to achieve TACACS+ data security through IPSec.

1. TACACS+/application initiates TCP connection with the TACACS+ server in the form of first TCP-SYN packet.
2. SYN packet is routed to TUN interface where it's directly read by the IpsecMgr in local context.
3. IpsecMgr sends the TCP-SYN packet to the first instance of NPUSIM for ACL match.
   a. If ACL entry matches with the TCP-SYN packet, it sends the packet back to the IpsecMgr/local.
   b. If the packet doesn't match the ACL entry, NPUSIM sends the packet to the local management interface bypassing the need to encrypt the packet.
4. IpsecMgr/local context receives the packet from NPUSIM after ACL match. It triggers the formation of IPSec tunnel with its peer by exchanging the IKE-INIT/IKE-AUTH packets using local raw socket created in local context.
5. The first TCP-SYN packet is dropped in the IpsecMgr/local after triggering the IPSec tunnel creation.
6. TACACS+/application sends another TCP-SYN packet and steps 2-3b are repeated.
7. When IpsecMgr receives the second TCP-SYN packet after ACL match from NPUSIM and the tunnel is already established, it encrypts the TCP-SYN packet and sends out through ESP raw socket created in the local context by the IpsecMgr/local.

8. IpsecMgr also listens for any ESP packets coming from ESP raw sockets in the local context via management ports.
9. On receiving any ESP packets, IpsecMgr/local sends ESP packet to NPUSIM for any SA flow processing.
10. If the SA flow matches in the NPUSIM, the ESP packet is sent to the IpsecMgr/local which does the decryption of the packet.
11. This packet could be TCP-SYN-ACK which could be the response of the second TCP-SYN packet sent from TACACS+ client to the TACACS+ server.
12. The decrypted packet is sent back to the same TUN interface from where it's sent back to the TACACS+/application.
13. The 2-way communication will be established, by the TACACS+/application which sends out the TCP-ACK packet. The above steps will be repeated to achieve the data security for all subsequent packets.

Recovery

IPsec tunnels are established between TACACS+ client on Active and the TACACS+ server application. There's no IPsec tunnel established between Standby and TACACS+ server. In usual scenario, IPsec endpoints exchange informational (heartbeat) messages to check the health of the IPsec tunnels. If an Active VNF goes down, IPSec endpoint at the TACACS+ server detects dead peer detection (DPD) of the IPsec endpoint on the Active VNF where DPD timeout is also configurable. DPD triggers the clearance of the tunnels on the TACACS+ server side. Once the Standby VNF comes back as Active and TACACS+ application starts to exchange data with the TACACS+ server application, a new IPsec tunnel is established between new Active VNF and the TACACS+ server.

Limitation

Following are the known limitations of the feature:

- TACACS+ using IPv6 is not supported with IPSec that uses IPv6 tunnel endpoints. However, without IPSec, TACACS+ using IPv6 is supported. Also, TACACS+ using IPv4 is supported with and without IPSec using IPv4 tunnel endpoints.
- The crypto maps in the local context must be pre-configured to be part of Day-0/Day-1 configuration. That is, crypto maps in local context, if any, must be configured before crypto maps are configured in any other context.

Configuring TACACS+ over IPSec

This section describes how to configure the TACACS+ over IPSec feature.

Configuring the feature involves the following steps:

1. Configuring TACACS+ Configuration Mode.
2. Provisioning TACACS+ with IPSec.
3. Provisioning TACACS+ with IPSec in Tunnel Mode.

4. Provisioning TACACS+ with IPSec in Transport Mode

Configuring TACACS+ Configuration Mode

Configuration to provision TACACS+ on StarOS/VNF remains the same as was done in non-CUPS architecture. However, for tunnel establishment in "IPSec Tunnel Mode", it's mandatory to provision the **src-ip**. You must reserve one extra Source IP address (*src_ip*) for TACACS+ communication and secure its communication.

For tunnel establishment in "IPSec Transport Mode", there's no requirement to provision an extra **src-ip**. The management interface IP address is picked as the **src-ip**.

The following is a sample configuration:

```
configure
   context context_name
      tacacs mode
         server priority priority_number ip-address server_ip_address password
text_password src_ip
         accounting command
         authorization prompt
      #exit
   aaa tacacs+
end
```

Provisioning TACACS+ with IPSec

The following configuration ensures that all IKE/ESP packets are handled in the user-space IpsecMgr/local and not by the IpsecMgr of non-local context and underlying data-plane like VPP, IFtask, or NPU.

```
configure
   require crypto ikev1-acl software context context
   require crypto ikev2-acl software context context
end
```

Provisioning TACACS+ with IPSec in Tunnel Mode

The following example configuration creates crypto map in the local context in Tunnel mode wherein **209.165.201.1** and **209.165.200.225** is assumed as the TACACS+ server and client IP address respectively.

Note Currently, Tunnel mode is supported only in IKEv2.

```
configure
  context local
    ip access-list foo
      permit ip 209.165.200.225 1 0.0.0.0 209.165.201.1 0.0.0.0
    #exit
    ipsec transform-set B-foo
      group 14
    #exit
    ikev2-ikesa transform-set ikesa-foo
      group 14
```

```
#exit
crypto map foo ikev2-ipv4
  match address foo
  authentication local pre-shared-key encrypted key EncryptedKey1
  authentication remote pre-shared-key encrypted key EncryptedKey2
  ikev2-ikesa max-retransmission 3
  ikev2-ikesa retransmission-timeout 2000
  ikev2-ikesa transform-set list ikesa-foo
  ikev2-ikesa rekey
  payload foo-sa0 match ipv4
    ipsec transform-set list B-foo
    rekey keepalive
  #exit
  peer 209.165.200.226
  ikev2-ikesa policy error-notification
#exit
interface local1
  ip address 209.165.200.227 255.255.255.224
  ipv6 address 2001:420:2c7f:f620::83/64 secondary
  crypto-map foo
#exit
```

Provisioning TACACS+ with IPSec in Transport Mode

The following example configuration creates crypto map in the local context in Transport mode wherein **209.165.200.229** is assumed as the TACACS+ server IP address.

Note Currently, Transport mode is supported only in IKEv1.

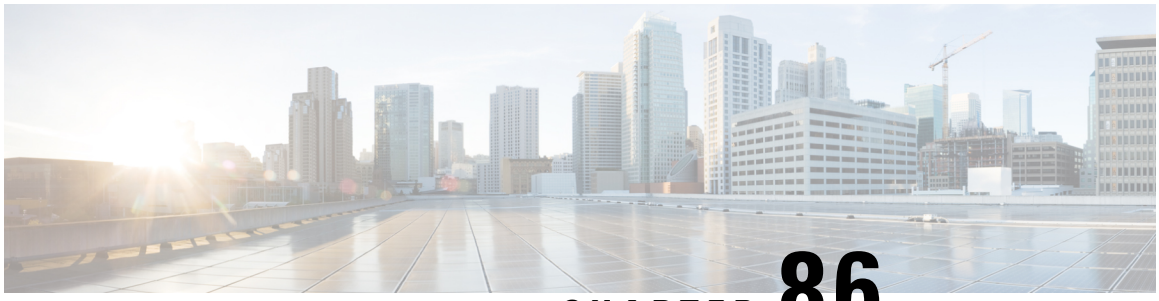
```
configure
  context local
    ip access-list foo
      permit tcp 209.165.200.228 0.0.0.0 209.165.200.229 0.0.0.0
    #exit
    ip routing shared-subnet
    ikev1 keepalive dpd interval 3600 timeout 10 num-retry 3
    crypto ipsec transform-set A-foo esp hmac sha1-96 cipher aes-cbc-128
      mode transport
    #exit
    ikev1 policy 1
    #exit
    crypto map foo ipsec-ikev1
      match address foo
      set peer 209.165.200.229
      set ikev1 encrypted preshared-key EncryptedKey1
      set pfs group2
      set transform-set A-foo
    #exit
    interface local1
      ip address 209.165.200.228 255.255.255.224
      ipv6 address 2001:420:2c7f:f620::84/64 secondary
      crypto-map foo
    #exit
```

Monitoring and Troubleshooting

Show Commands and Outputs

The following show CLI commands are available in support of this feature.

- **show crypto map**
- **show crypto ikev2-ikesa security-associations** *summary*
- **show crypto ikev1 security-associations** *summary*
- **show crypto statistics**
- **show crypto ipsec security-associations** *summary*



CHAPTER 86

Tariff Time Support

- Revision History, on page 753
- Feature Description, on page 753

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The Tarrif switch time functionality is applied when a subscriber switches form one tarrif plan to another.

The Tariff-Time-Change AVP is used to determine the tariff switch time, and the Monitoring-Time IE is used to support the Tarrif Time support functionality.

After a tariff timer expiry, the Gateway accumulates the usage separately in a charging bucket and continues to consume from the original quota value. At the time of next reporting, (Quota exhausted or another control events) the Gateway will report both usages (before and after tarrif time change) for the same Charging Bucket.

The first reporting of this charging-bucket will have the Reporting-Reason: Tariff-Time-Change, and the second bucket will contain the last reporting reason, and the quota usage after the tariff-timer expiry.

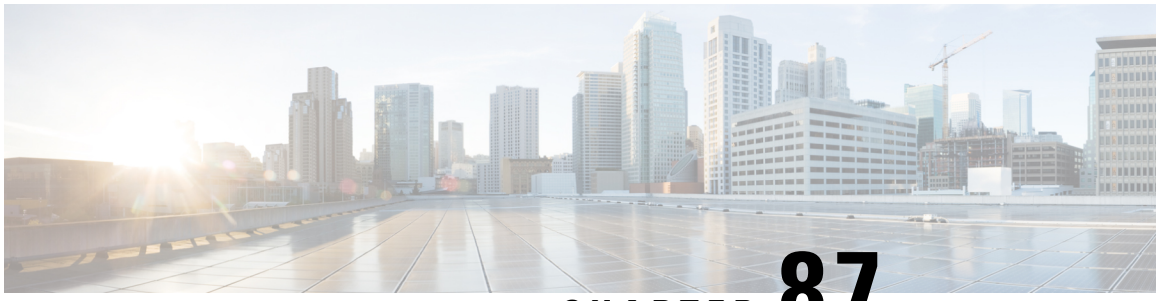
The data traffic usage can be split into resource usage before a tariff switch and resources used after a tariff switch. The Tariff-Change-Usage AVP is used within the Used-Service-Units AVP to distinguish reported usage before and after the tariff time change.

Limitations

Following are the known limitations of this feature:

- Only one tariff time per RG/Service ID combination is supported.

- Allocation of different quota before and after tariff time change is not supported. This functionality is not in compliance with the 3GPP stanadards.



CHAPTER 87

UP Call Summary Log

- Revision History, on page 755
- Feature Description, on page 755
- How it Works, on page 756
- Interdependencies, on page 758
- Limitations and Restrictions, on page 759
- Configuring Call Summary Log in UP, on page 759
- Monitoring and Troubleshooting, on page 760

Revision History

Revision Details	Release
First introduced.	21.24.1

Feature Description

The User Plane Call Summary Log (CSL) is a mechanism to report the following parameters to external log collection server:

- Session establishment
- Session Modification
- Session Deletion
- Usage Reporting

The system uses the CSL record to analyze and debug subscriber call.

This feature supports the following functionality:

- Support for UP CSL for Sxb and Sxab interface (For Pure-P, and Collapsed calls)
- The system stores the CSL records only in CSV format.
- The Sessmgr_u buffers the CSL records for defined time interval of max 30s.

How it Works

The interface between the UP and log collection server is based on SFTP. Each record is in the form of comma-separated ASCII values (CSV record). The UP sends one ASCII formatted CSV record per line. The system stores the CSV records in file and compresses the file before sending to external collection server. The CSV records can't be older than 15 minutes. So, the file needs to be SFTed to external collection server at least once in 15 minutes. The transfer of CSV record file between UP and collection server is either PULL or PUSH model. In case of PULL model, the external collection server is responsible for SFTP with UP. In case of PUSH model, UP is responsible for sending the CSV record file to external collection server. The file transfer happens based on the configured PUSH timer interval.

The following events trigger CSL record.

Event	Description
1	Session Establishment Request/Response
2	Session Modification Request/Response
3	Session Deletion Request/Response
4	Usage report Request/Response

The CSL record includes the following information in the CSV format:

Number	Description	Format Example
1	UP CSL Record No	Integer <proclet-type>|<instance-id>|<RTT-record-#>
2	UP Version No	Integer Version 1 in v21.24.0
3	Procedure No	PFCP IEs 29.244(Table 7.3-1: Message Types)
4	UP Name	Host Name of the Chassis
5	Procedure Start Time	Time in UTC, (to ms accuracy)
6	Procedure End Time	Time in UTC (to ms accuracy)
7	ASR5K CallID	Internal CallID 376efb10
8	Sx-PFCP Remote SEID	
9	Interface Type	
10	Reserved	
11	IMSI	Integer (15 digits) Example: [311480076488840]
12	MSISDN	Integer Example: [19728256305]

Number	Description	Format Example
13	IMEISV	Integer (16 Digits) Example: [9900028823793406]
14	RAT	IPv6 Address
15	SGW TEID (FARID, RTEID)	Tunnel Identifier for Peer Example: 1,0x084BC005|2,0x084BC01|3,0x084BC010|
16	PGW TEID (PDR ID, FTEID)	Tunnel Identifier for UP Example: 1,0x084BC005|2,0x084BC010|3,0x084BC010|
17	APN	String Example: [vzwims.mnc311.mcc480.3gppnetwork.org]
18	IPv4 Address	IPv4 Address UE assigned IPv4 address
19	IPv6 Address	IPv6 Address UE assigned IPv6 prefix/address
24	Uplink AMBR	Integer (0-4 Billion) In Kbps Example: [0-4294967295]
25	Downlink AMBR	Integer (0-4 Billion) In Kbps Example: [0-4294967295]
26	Uplink MBR	Integer (0-4 Billion) In Gbps. MBR. (QER ID, MBR) Example: 1,1234 | 2,3456 |3, 567
27	Downlink MBR	Integer (0-4 Billion) In Gbps. MBR. (QER ID, MBR)
	Uplink GBR	Integer (0-4 Billion) (QER ID, GBR)
	Downlink GBR	Integer (0-4 Billion)
	Sx Response Value	(Cause, Offending IE) 1,0 OR 64,44 Request/Acceptance/Rejection Cause, Example: [1-255] 1 thru 6
	PFCP Session Establishment Request/Response	Create PDR 1|2|3|4 Create FAR 1|2 Create QER 1|2 Create URR 1|2|3|4 Create TE 1|2

Number	Description	Format Example
	PFCP Session Modification Request/Response	Create PDR 5|6 Update PDR 3|4 Remove PDR 1|2 Create FAR 1|2 Update FAR. (RTEIDxxxx, Apply action) Create QER 1|2 Create URR 1|2|3|4 Create TE 1|2 Update TE 1|2
	PFCP Session Deletion Request/Response	Remove PDR 5|6 Remove FAR 7|8 Remove URR 1|2 Remove QER. Remove TE
	PFCP Session Report Request/Response	Report Type? (DLDR|USAR....|UISR)

Fault and Fault Reporting

The sessmgr provides a warning message in case of failure to post CSLs to cdrmod or any buffering issues due to memory allocation. There are no SNMP traps defined to report faults by Sessmgr. The Cdrmod separately reports the fault issue in case of issue in storing UP CSL record files in RAM.

Redundancy

Both Session recovery and ICSR are supported for UP CSL. After Sessmgr recovery, UP CSL continues, if UP CSL is enabled for the subscriber. Similarly, for ICSR calls, UP CSL continues, if UP CSL is enabled for the subscriber. During Session recovery and ICSR, the locally buffered Sessmgr CSL records are lost.

Session recovery at cdrmod requires minimal support because CSL record files are stored either using RAMFS or using Hard disk and are available across recovery. For ICSR, UP CSL record file needs to be transferred from old active chassis to new active chassis.

Interdependencies

Following CDRMOD functionality is required in support of this feature:

- New CDR Module type to support UP CSL records.
- Storing of UP CSL records using RAMFS

- Compression of UP CSL record file.
- Fault reporting
- SNMP Trap generation
- Stats /Bulkstat support
- Session Recovery and ICSR

Limitations and Restrictions

To enable this feature, CDRMOD, UP service configuration, and SFTP configuration is required. The CDRMOD configuration is required to setup the CDRMOD with necessary configuration parameters like CDRMOD module type, compression method, storage method and so on. The UP-service configuration is required to enable the reporting of UP CSL. The SFTP configuration is required to transfer the UP CSL record file from chassis to external collection server.

Configuring Call Summary Log in UP

Enabling/Disabling the CSL

Use the following configuration to enable or disable the reporting of UP event records to log.

```
configure
   context context_name
      apn apn_name
         [ no | default ] reporting-action up-event-record
         end
```

NOTES:

- **reporting-action:** Configures the event reporting
- **up-event-record:** Enables the reporting of event records. By default, reporting of event records is disabled

UP Service Configuration

Use the following as the UP-service configuration:

```
session-event-module
 file name evt-repo rotation volume 2097152 rotation time 30 compression gzip
 event use-harddisk
 event remove-file-after-transfer
 event transfer-mode push primary url  sftp://xxxxxxxx@xx.xx.xxx.xxx/tmp/  via local-
context
 event push-interval 30
```

Monitoring and Troubleshooting

At Sessmgr level, use the **show subs sgw-only full** CLI command to know whether UP CSL is enabled or not. You can also enable the Session manager warning message to know in case there is issue in reporting events at Sessmgr.

The CDRMOD uses separate CLI and/or SNMP Trap and/or Warring/Debug messages to aid troubleshooting of CDRMOD issue related to UP CSL.

Statistics

The following CLI command is available in support of the feature.

show up-event-record statistics interface-type [sxb | sxab | n4]

NOTES:

- **up-event-record**: Displays the number of event records.
- **statistics**: Displays the statistics for event records.
- **interface-type**: Displays the event records for interface type.

Show Command Outputs

Following is the sample output for **show user-plane-service up-event-record statistics interface sxb** CLI:

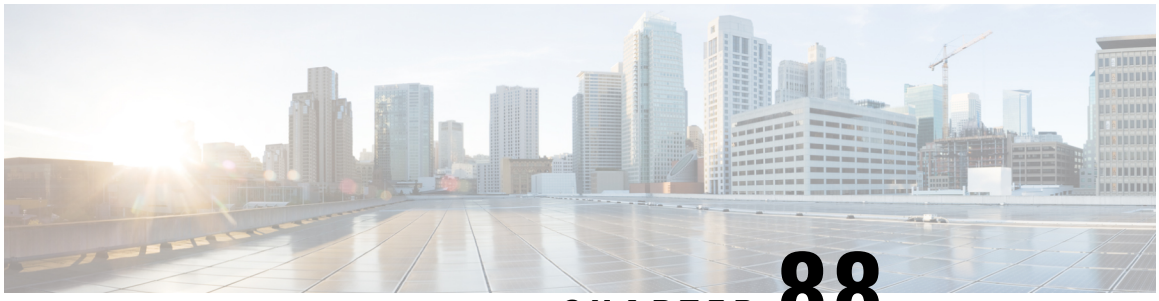
```
Number of event records: 80
Number of event records for sx procedures: 50
   PFCP Session Establishment procedure: 10
   PFCP Session Modification procedure: 20
   PFCP Session Deletion procedure: 10
   PFCP Session Report procedure: 100
```

Following is the sample output for **show config** or **show config verbose** when reporting of event records is enabled.

```
config
  context <>
    apn <>
      …
      reporting-action up-event-record
```

Following is the sample output for **show config verbose** when reporting event records is disabled.

```
config
  context <>
    apn <>
      …
      no reporting-action up-event-record
```



CHAPTER 88

URL Blockedlisting

- Revision History, on page 761
- Feature Description, on page 761
- How it Works, on page 761
- Configuring URL Blockedlisting, on page 763
- Monitoring and Troubleshooting, on page 764

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The URL blockedlisting feature regulates the subscribers access to view or download content from websites whose URL or URI has been blockedlisted. It uses a database that records a list of URLs that indicates if the detected URL is categorized to be blocked or not.

How it Works

To enable the URL blockedlisting feature on User Plane (UP), URL blockedlisting database should be present with a name “optblk.bin” under flash, or SFTP or under its sub-directory. This database directory path needs to be configured on user-plane, after user-plane services are brought up.

HTTP Analyzer must be enabled for URL blockedlisting. The HTTP analyzer extracts URL information from the incoming HTTP request data packet. Extracted URL content is compared with the URL Blockedlisting database. Once the incoming HTTP data packet’s URL matches with the database URL entry, that URL is treated as blockedlisted URL and one of the following actions takes place on that HTTP packet.

- Termination of flow
- Packet is discarded

The URL blockedlisting configurations must be configured on Control Plane (CP), Rulebase configuration under Active Charging Service. Additionally, two URL blockedlisting methods – Exact and Generic, are supported at Active Charging Service-level configuration, on CP. These CLI configurations are pushed to UP through PFD mechanism, during Sx association procedure, to the CP.

Important Blockedlisting database(s) are provided by – IWF (Internet Watch Foundation) and NCMEC (National Center for Missing and Exploited Children). The ASR5500, CUPS UP always receives the blockedlisting DB in Optimized Format (optimized blockedlisting DB format).

URL Blockedlisting Database Upgrade

URL database upgrade is supported in 2 ways:

- Timer-based upgrade or Auto upgrade
- CLI-based upgrade or Manual upgrade

Timer-based or Auto-upgrade

After the database is loaded on the chassis for the first time, a timer, for a duration of 5 minutes, is started. This process is started to auto upgrade the database.

If at the expiry of the timer, a valid database with higher version is available at the directory path, then database upgrade procedure is initiated, and a newer version of the database is loaded on the UP chassis.

To upgrade a URL blockedlisting database, a higher version of valid URL Blockedlisting database with name “optblk_f.bin” should be present at same directory as that of current database “optblk.bin”.

After the database is upgraded successfully, the earlier “optblk.bin” file gets renamed as “optblk_0.bin” and “optblk_f.bin” file gets renamed as “optblk.bin”. Here, “optblk_0.bin” file is treated as a backup file of older database.

If one more upgrade is performed, then “optblk_0.bin” file will be renamed as “optblk_1.bin” file and current “optblk.bin” will get renamed as “optblk_0.bin”, and so on.

The number of backup files to be stored in the database can be configured using the **max-versions** CLI on UP.

CLI-based or Manual Upgrade

In this upgrade method, the CLI command - **upgrade url-blacklisting database**, upgrades the current database to a newer version.

Limitations

In this release, session recovery and user-plane redundancy support is not fully qualified.

Configuring URL Blockedlisting

Loading URL Blockedlisting Database on UP

Use the following configuration to load URL blockedlisting database on UP.

In releases prior to StarOS 21.26:

```
configure
  url-blacklisting database directory path database_directory_path
  url-blacklisting database max-versions max_version_value
  end
```

From StarOS 21.26 and later releases:

```
configure
  url-blockedlisting database directory path database_directory_path
  url-blockedlisting database max-versions max_version_value
  end
```

NOTES:

- **database directory path**: Configures the database directory path.

  The *database_directory_path* is a string of size 1 to 255.

- **max-versions**: Configures the maximum database upgrade versions.

  The *max_version_value* is an integer from 0 to 3.

Configuration to Enable URL Blockedlisting

Use the following configuration to enable URL blockedlisting feature on Control Plane.

In releases prior to CUPS 21.26:

```
configure
  require active-charging service_name
    url-blacklisting match-method [ exact | generic ]
    rulebase rulebase_name
      url-blacklisting action [ discard | terminate-flow ]
      end
```

From CUPS 21.26 and later releases:

```
configure
  require active-charging service_name
    url-blockedlisting match-method [ exact | generic ]
    rulebase rulebase_name
      url-blockedlisting action [ discard | terminate-flow ]
      end
```

NOTES:

- **match-method [exact | generic]**: Specifies the match method used for URL blockedlisting.

exact: URL Blockedlisting perform an exact-match of URL.

generic: URL Blockedlisting perform generic-match of URL.

- **url-blockedlisting action [discard | terminate-flow]**

discard: Discards the HTTP packet received.

terminate-flow: Terminates the flow of the HTTP packet received.

URL Blockedlisting Database Upgrade

Use the following command to upgrade the URL Blockedlisting Database.

In releases prior to CUPS 21.26:

```
upgrade url-blacklisting database
```

From CUPS 21.26 and later releases:

```
upgrade url-blockedlisting database
```

Note This CLI is used for manual upgrade of URL Blockedlisting database. File optblk_f.bin must be present in order to upgrade URL Blockedlisting database.

Monitoring and Troubleshooting

This section provides information regarding the CLI command available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show user-plane-service url-blacklisting database

The following fields are displayed in support of this feature:

- URL Blacklisting Static Rating Databases:
  - Last Upgrade Status
  - Path
    - Database Status
    - Number of URLs in DB
    - Type
    - Version
    - Creation Time

- Hostname
- Comment
- Last Access Time
- Last Modification Time
- Last Status Change Time

show user-plane-service url-blacklisting database url *database_directory_path*

The following fields are displayed in support of this feature:

- URL Blacklisting Static Rating Databases:
  - Last Upgrade Status
  - Path
    - Database Status
    - Number of URLs in DB
    - Type
    - Version
    - Creation Time
    - Hostname
    - Comment
    - Last Access Time
    - Last Modification Time
    - Last Status Change Time

show user-plane-service url-blacklisting database facility sessmgr all

The following fields are displayed in support of this feature:

- URL-Blacklisting SessMgr Instance Based Database Configuration
  - SessMgr Instance
  - BL DB Load Status
  - BL DB Version
  - Number of URLs
  - Checksum

show user-plane-service inline-services info

The following fields are displayed in support of this feature:

- URL-Blacklisting: Enabled
  - URL-Blacklisting Match-method: Generic

show user-plane-service rulebase name *rulebase_name*

The following fields are displayed in support of this feature:

- URL-Blacklisting Action
- URL-Blacklisting Content ID

show user-plane-service inline-services url-blockedlisting statistics

The following are displayed in support of this feature:

- Cumulative URL-Blockedlisting Statistics
  - Blockedlisting URL hits
  - Blockedlisting URL misses
  - Total rulebases matched

show user-plane-service inline-services url-blacklisting statistics rulebase name *rulebase_name*

The following fields are displayed in support of this feature:

- Rulebase Name
  - URL-Blacklisting Statistics
  - Blacklisted URL hits
  - Blacklisted URL misses
- Total rulebases matched

Bulk Statistics

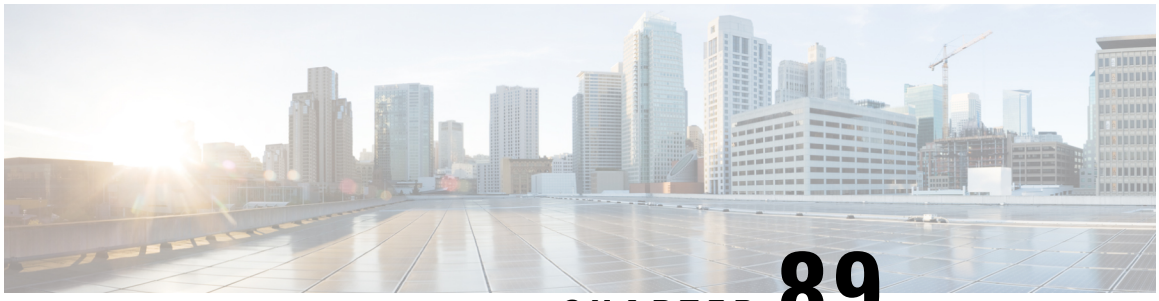
The following bulk statistics are added to the System schema in support of URL Blacklisting feature:

- **url-blacklisting-hits**: Indicated the total number of URLs blacklisted.
- **url-blacklisting-misses**: Indicated the total number blacklisted URLs missed.

SNMP Traps

The following SNMP trap are added in support of this feature:

- **BLDBError**: Specifies the blacklisting OPTBLDB file error displayed with an error code.
- **BLDBErrorClear**: Specifies the blacklisting OPTBLDB file error removed.
- **BLDBUpgradeError**: Specifies the blacklisting OPTBLDB file error displayed with an error code.
- **BLDBUpgradeErrorClear**: Specifies the Blacklisting OPTBLDB file error removed.



CHAPTER 89

User Plane Selection based on TAC Range

- Revision History, on page 769
- Feature Description, on page 769
- How It Works, on page 769
- Configuring User Plane Selection based on TAC Range, on page 771
- Configuring Tracking Area Code Profile, on page 772
- Configuring Routing Area Code Profile, on page 773

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
With this release, support is added for TAC/RAC profile configuration.	21.25
First introduced	Pre 21.24

Feature Description

With this feature, User Plane group can be selected based on Access Point Name (APN). The ability to configure Tracking Area Code (TAC) range, in rule combinations in virtual APN selection, helps in giving more flexible network design for location-based User Plane selection for edge computing and other services.

With 21.25 and later releases, support is added to configure TAC and Routing Area Code (RAC) profile in the Control Plane node. Using this feature, it is now possible to select APN based on discrete values of TAC/RAC profile instead of range.

How It Works

In non-CUPS architecture, Virtual APN selection is based on the following parameters:

- Subscriber IP
- Access-gw-address
- Bearer-access
- cc-behavior
- cc-profile
- domain
- mcc
- msisdn-range
- pdp-type
- rat-type
- roaming-mode
- serv-gw-plmnid

In CUPS architecture, Virtual APN selection is based on Tracking Area Code range with other options, such as cc-profile or mcc/mnc.

To support this feature:

- A new CLI keyword is introduced to accommodate new parameter.
- During call processing, incoming tracking area code is compared with the configured tracking area code range to determine the Virtual APN.

The Tracking Area Code based Virtual APN selection:

- Supports at least 30 tracking-area-code-range configured for Virtual APN.
- Supports overlapping ranges (subset or superset). Duplicate of tracking-area-code-range is not allowed for different priority.
- Selects a Virtual APN based on CLI configuration and User Plane is selected based on Virtual APN for a new call based on the tracking-area-code for that UE.
- Supports a combination of tracking-area-code-range and cc-profile in same priority.

Virtual APN functionality includes storing all the Virtual APN selection rules per real/Gn APN. Every rule has multiple criteria. Rule is identified by preference number. The list of APNs are stored and within APN a rule is identified using preference number.

New parameter has been introduced to pass Tracking Area Code, received in CSReq (TAI).

Limitations

Following are the known limitations and restriction of this feature.

- New configuration with multiple selection criteria in Virtual APN selection does not work with older builds/releases. User should have separate copies of the configuration for older builds/releases.

- Modify operation on the Virtual APN rule is not supported. User should delete the existing rule and add new rule to achieve modify operation.
- If same option is provided multiple times in the same rule, then the value of later option is considered for selection.
- Total number of Virtual APN rules added across all APNs is limited to 2048. This limitation exists in non-CUPS architecture.
- Upto 1000 TAC/RAC profiles can be configured. Memory usage is based on the number of profiles configured.
- The maximum number of TAC/RAC discrete values supported in a profile are 100. Memory usage is fixed per profile.
- TAC/RAC range or discrete values can overlap between profiles to support maintenance activities like split existing profile or others.
- This is Day-0 and Day-1 configuration.
- Multiple profiles can be associated with an APN.
- There are no changes in existing IP pool functionality.
- There is no specific impact on ICSR or Multi-Sx configurations.
- There is no Service Area Code (SAC) support.
- Pure-S calls aren't supported.
- UP selection requirements are handled in multi-UP group support features.

Configuring User Plane Selection based on TAC Range

This section provides information about CLI commands available in support of this feature.

Configuring Tracking Area Code Range

Use the following CLI commands to configure APN for Tracking Area Code range in Control Plane node.

```
configure
  context context_name
    apn apn_name
     virtual-apn preference preference apn apn_name tracking-area-code-range
 tac_range
      end
```

NOTES:

- **tracking-area-code-range** *tac_range*: Configures APN for Tracking Area Code range. The *tac_range* is an integer value ranging from 0 to 65535.

Verifying the Tracking Area Code Range Configuration

Use the following CLI commands to verify if the feature is enabled and the range that is configured for Tracking Area Code.

- **show configuration apn** *apn_name*
- **show apn name** *apn_name*

Configuring Tracking Area Code Profile

From 21.25 and later releases, Tracking Area Code profile can be configured in the Control Plane node. Using this feature, it is now possible to select APN based on discrete values of TAC instead of only range.

The following CLI commands are used to configure Tracking Area Code profile with discrete values and range.

```
configure
   context context_name
      tac-profile tac_profile_name
         tac range X to Y
         tac value
```

NOTES:

- **tac-profile** *tac_profile*: Configures APN for Tracking Area Code profile. The *tac_profile* is any range or discrete integer value ranging from 0 to 65535.
- The number of discrete TAC values supported per CLI command is 16.

Associating TAC Profile with APN

Use the following configuration to associate TAC profile with APN:

```
configure
   context context_name
      apn apn_name
         virtual-apn preference preference apn apn_name tac-profile tac_profile
         end
```

Verifying the Tracking Area Code Profile Configuration

Use the following CLI commands to verify if the feature is enabled and the range that is configured for Tracking Area Code profile.

- **show configuration apn** *apn_name*
- **show apn name** *apn_name*
- **show rule definition** *tac_profile*

Configuring Routing Area Code Profile

From 21.25 and later releases, Routing Area Code profile can be configured in the Control Plane node. Using this feature, it is now possible to select APN based on discrete values of RAC profile instead of range.

The following CLI commands are used to configure Routing Area Code profile with discrete values.

```
configure
   context context_name
      rac-profile rac_profile_name
         rac range X to Y
         rac value
```

NOTES:

- **routing-area-code-profile** *rac_profile*: Configures APN for Routing Area Code profile. The *rac_profile* is any range or discrete integer value ranging from 0 to 255.
- The number of RAC profile values supported is upto 16.

Associating RAC Profile with APN

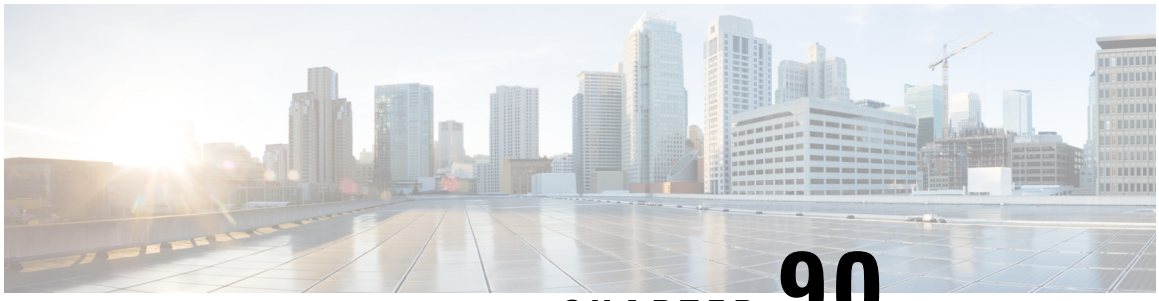
Use the following configuration to associate TAC profile with APN:

```
configure
   context context_name
      apn apn_name
         virtual-apn preference preference apn apn_name
routing-area-code-profile rac_profile
         end
```

Verifying the Routing Area Code Profile Configuration

Use the following CLI commands to verify if the feature is enabled and the range that is configured for Routing Area Code profile.

- **show configuration apn** *apn_name*
- **show apn name** *apn_name*
- **show rule definition** *rac_profile*



CHAPTER 90

User Plane Node Bring Down Procedure

- Revision History, on page 775
- Feature Description, on page 775
- Preconditions, on page 776
- How it Works, on page 776
- Limitations and Considerations, on page 777
- Configuring UP Node Bring Down Procedure, on page 778
- Monitoring and Troubleshooting, on page 778

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The User Plane Node Bring-Down Procedure is a Maintenance Operating Procedure (MoP) that outlines the steps to bring down a specific User Plane (UP) node for maintenance operations. The purpose of this procedure is to disable specific UP node on Control Plane (CP) node while node selection takes place for new incoming sessions.

This feature provides the following functionality:

- Configuration to mark a specific UP unavailable for any new session.
- Option to delete idle subscribers.

Preconditions

Following are the preconditions to bring down the UP node for maintenance operation:

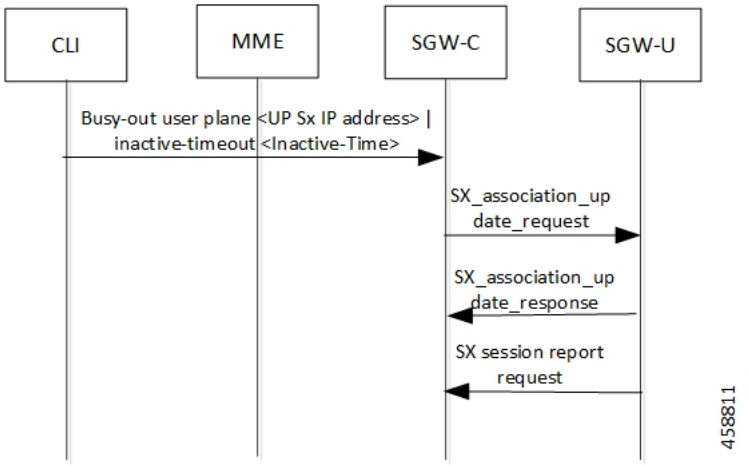
- The UP and CP nodes must be associated with each other and that the calls land on a specific UP.
- You want to disable the specific UP node for maintenance, from being selected for new incoming calls, and clear existing idle users on that UP node.
- To avoid any call loss, there's a provision of another UP node in the same CP group. If there's only one UP in the group and we disable it for maintenance, the CP rejects the new incoming sessions with disconnect reason as "user-plane-info-not-available".

How it Works

Call Flow

UP Selection when a UP is Marked Busy Out

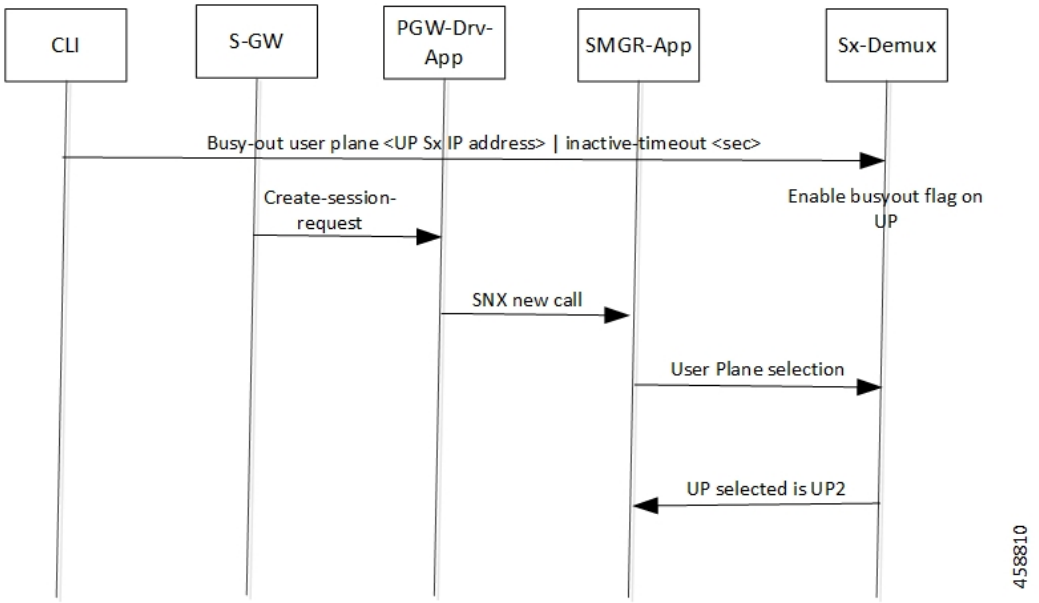
The following call flow describes about the UP selection for Pure-P and Collapse call when some UP are marked "busy-out" through the CLI command. Similarly, the UP selection for Pure-S call also takes place.



Steps	Description
1.	User configures the busy-out configuration from the CP to "busy-out" a specific UP. The inactive time value is optional in the configuration.
2.	The association state of UP is seen as "B". An association update request message is sent to User Plane with "busy-out" inactive time value if inactive time value is configured.

UP Clear Idle Subscribers based on Busy Out Inactivity Timeout

The following call flow describes how inactive sessions (Pure-S call) are cleared on UP when inactive-time is configured in “busy-out” CLI. Other call types also work similarly.



In continuation to the previous call flow, this call flow describes about the clearing of the idle subscriber based on the busy-out inactivity- timeout:

Step	Description
1.	The calls are cleared if they remain idle for the period equivalent to the inactivity time value. The UP can be pulled back to the associated state by configuring the “no” form of the same CLI command.

Limitations and Considerations

Following are the known limitations of this feature:

- The multi PDN calls for same UE can fall to different UPs.
- The “busy-out” configuration is done on all Active and Standby CPs.
- Currently, the new IP Pool isn’t added for the specified UP or the belonging UP group so that some of the IP chunks doesn’t get assigned to this UP, leading to loss in capacity.
- All CPs must have the same configuration for “busy-out”. Else, UP uses the latest configuration value triggered from any one of the CPs. Similarly, when you do the “no busy-out” on any of the CPs, UP comes out of “busy-out”.
- To block the UP completely, “busy-out” both its IPv4 and IPv6 addresses using two separate CLIs.
- Currently, there’s a spike in CPU usage when idle timeout triggers for a huge number of calls. The CPU usage eventually goes down when all the calls get cleared out.

Configuring UP Node Bring Down Procedure

The MoP is applicable only when CPs and UPs are on the same software version where the support of the following “busy-out” CLI is available.

```
configure
  busy-out user-plane { ipv4-address ipv4_address | ipv6-address ipv6_address
 } [ inactive-timeout inactive_time ]
  end
```

You must enable the configuration on CP to make a UP unavailable for new sessions and clear the idle sessions which exceeds the time configured in “inactive-timeout”. The *ipv4_address*/*ipv6_address* is the IPv4 or IPv6 address of the UP. “Inactive-Time” is configured in seconds.

If you don’t configure the inactive timeout, then the behavior for the idle session remains unchanged.

NOTES:

- Existing **clear subscribers saegw-only uplane-address** *ip_address* **no-select-up** CLI command is not extended or reused for “busy-out” logic as it is exec-level CLI which is not fit in the scenario of session-recovery. There is also no other way to roll back the up-selection logic without reassociating the UP using this CLI.

Note When the **clear subscribers** command is executed on UP, CP will not be informed and will consider the sessions as running.

- After “busy-out” CLI command is executed, the UP is removed from UP selection. Existing calls continue to function as usual. No extra operation is performed to the assigned IP pool chunk.
- If you want to bring the same UP after upgrade to handle new calls, you must undo the configuration by executing the **no busy-out user-plane** { **ipv4-address** *ipv4_address* | **ipv6-address** *ipv6_address* } CLI command.
- After “inactive-timeout” is configured for UP in “busy-out” CLI, calls get cleared and some of the assigned pool chunks to this UP gets reclaimed if the other UPs sharing the same IP Pool reach threshold of approximately 70% of the chunks allocated.

Monitoring and Troubleshooting

Following are the CLI commands available in support of this feature.

Show Commands and Outputs

show sx peers

The output of this CLI command has been enhanced to include the following new association state:

- Busy-Out: Indicates that the given UP is undergoing “busy-out” operations and is not available for new calls.

show sx peers wide

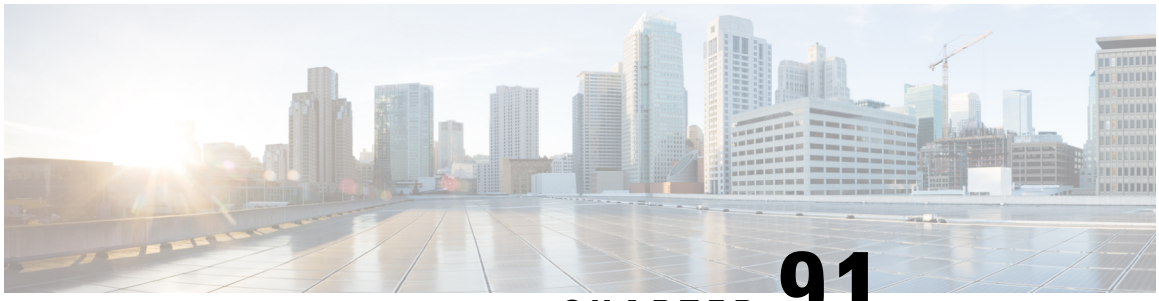
The output of this CLI command has been enhanced to include the following new fields:

- Last Busy-Out Time: Indicates the time at which UP last stayed in “busy-out” state.
- Last Busy-Out Clear Time: Indicates the time when “busy-out” state in UP was last cleared

Following is a sample output of **show sx peers** CLI command.

```
+---Node Type:    (C) - CPLANE       (U) - UPLANE
|
|+--Peer Mode:    (A) - Active       (S) - Standby
|
||+-Association   (i) - Idle         (I) - Initiated
||| State:        (A) - Associated   (R) - Releasing
|||               (X) - Released     (B) - Busy Out
|||
|||+Configuration (C) - Configured   (N) - Not Configured    (X) - Not Applicable
||||State:
||||
||||+IP Pool:     (E) -  Enable      (D) - Disable           (N) - Not Applicable
|||||
|||||
|||||
|||||  Sx Service
   No of
|||||  ID
  Restart
|||||  |                                                                          Recovery
     |    Current    Max       Peer
vvvvv  v       Group Name               Node ID                Peer ID       Timestamp
     v   Sessions  Sessions    State  LCI  OCI
----- ---- ------------------- ---------------------------- ---------- -------------------
 ---- --------- --------- -------- ---- ----
UABCE 20   default              209.165.200.225                   33554433
2021-04-14:01:25:32 0    0         1         NONE       X   X

Total Peers:    1

[local]qvpc-si# show sx peers wide

+---Node Type:    (C) - CPLANE       (U) - UPLANE
|
|+--Peer Mode:    (A) - Active       (S) - Standby
|
||+-Association   (i) - Idle         (I) - Initiated
||| State:        (A) - Associated   (R) - Releasing
|||               (X) - Released     (B) - Busy Out
|||
|||+Configuration (C) - Configured   (N) - Not Configured  (X) - Not Applicable
||||State:
||||
||||+IP Pool:     (E) -  Enable      (D) - Disable         (N) - Not Applicable
|||||
|||||+Push Config Status: (C) - Push Complete (P) - Push in Progress (X) - Not Applicable
||||||                     (E) - Push Error
||||||
||||||+Monitor State:    (U) - UP   (D) - DOWN  (N) - Not Applicable
```

```
|||||||
|||||||  ID
    Restart
|||||||  |                                                                                  Recovery
       |    Current    Max   Peer     Config    Auto-Config Config Push         Config Push
                  Last Busy-Out       Last Busy-Out
vvvvvvv  v       Group Name               Node ID                Peer ID       Timestamp
       v   Sessions  Session State    Failures  Success     Start Time          End Time
          LCI OCI   Time                Clear Time
------ ---- -------------------- ------------------------------ ---------- -------------------
 ---- --------- -------- -------  -------  ---------   ------------------- -------------------
 ---- ----  ------------------  --------------------
UAACECN 20   UP-Grp-1             209.165.200.225                   33554435
2021-05-10:12:41:03 0    1         1         NONE     0       0          2021-05-10:12:41:21
 2021-05-10:12:41:22   X   X   2021-05-10:12:42:50  2021-05-10:12:43:09

Total Peers:    1
```



CHAPTER 91

Virtual APN in CUPS

- Revision History, on page 781
- Feature Description, on page 781
- How It Works, on page 782
- Configuring Virtual APN in CUPS, on page 784

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
Support is added for virtual APN selection based on preference.	21.24.1
First introduced.	Pre 21.24

Feature Description

Access Point Name (APN) is a logical name referring to an external packet data network and/or to a specific service that the subscriber wishes to connect to.

Virtual APNs allow differentiated services within a single APN.

The Virtual APN feature allows a carrier to use a single APN to configure differentiated services. The APN that is supplied by the MME is evaluated by the P-GW along with multiple configurable parameters. Then, the P-GW selects an APN configuration that is based on the supplied APN and those configurable parameters.

APN configuration controls all aspects of a session at the P-GW. Different policies imply different APNs. However, after basic APN selection, internal reselection can occur based on the following parameters:

- Service name
- Subscriber type
- MCC-MNC of IMSI

- Domain name part of username (user@domain)
- S-GW Address

A call received on a particular APN can be redirected to another APN through a Virtual APN, based on a given criteria.

An APN received in the Create Session Request is called Gn APN, and the APN selected as part of a Virtual APN selection is called Gi APN.

Currently, the GGSN, P-GW, SAEGW non-CUPS products support Virtual APN selection that is based on the following modes:

- Local Configuration based
- Gx based
- RADIUS based
- Location based (for GGSN calls)

The P-GW/SAEGW deployed in CUPS mode also supports similar functionality to use the feature in network deployments.

How It Works

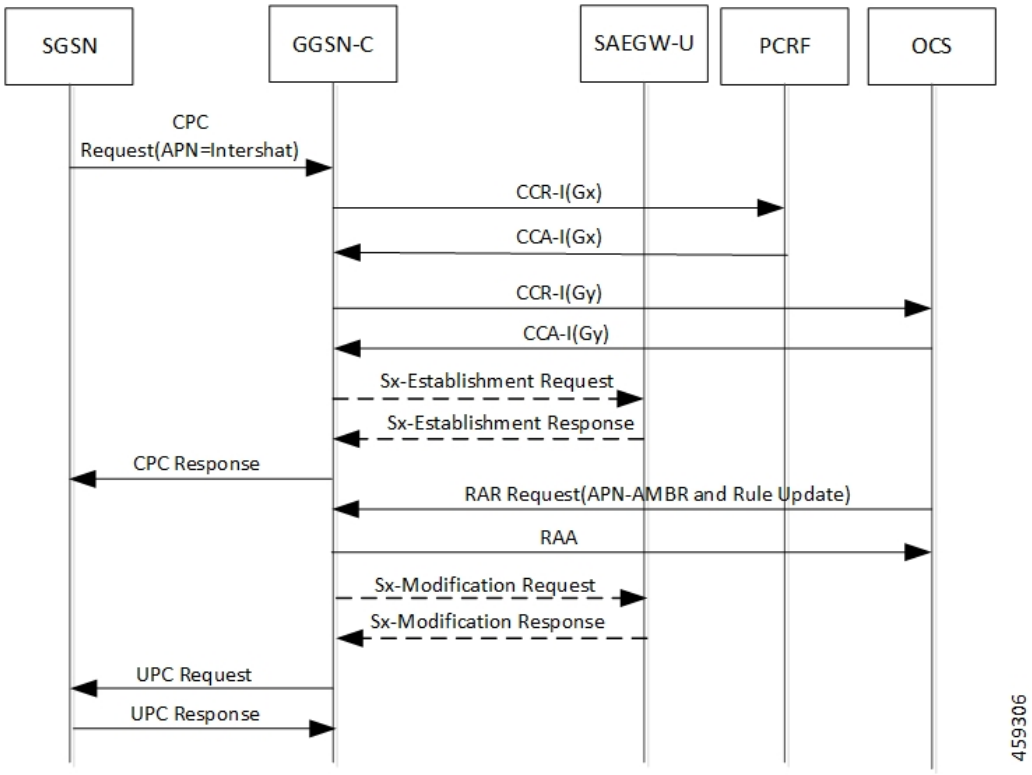
The Virtual APN feature is supported as a forward compatible to CUPS architecture-based P-GW/SAEGW nodes. Since the feature is being supported incrementally, following methods can be used to select Virtual APN for CUPS-based gateway nodes:

- Local Configuration based
- Gx based
- Location based (for GGSN calls)

Call Flow

The following call flow describes about the various steps for the VAPN Selection.

Figure 43: VAPN Selection



Following steps are performed in the event of the new calls:

Table 47: VAPN Selection call Flow

Steps	Description
1.	Extract roaming-mode, bearer-access, serv-gw-plmnid, pdp-type, along with all other required criteria from new create session request (or Create PDP Context Request).
2.	Extract the service name which is handling this call.
3.	Extract peer address which is sending the call.
4.	Pass all the parameters to the virtual APN selection code/algorithm.
5.	It's expected that either select some virtual APN or continue with GnAPN.

Limitations

Following are the known limitations and restrictions of this feature:

- If the same option is provided multiple times in the same rule, then later option value is considered for selection.
- New configuration with multiple options for virtual APN selection cannot be applied to older StarOS builds without this feature support. Therefore, you must keep separate copy of the older configuration (without multiple options selected) for older builds.
- Modify operation on the virtual APN rule is not supported. User must delete the existing rule and add new rule to achieve modify operation.
- A maximum of 2048 virtual APN rules can be added across all APNs.

Configuring Virtual APN in CUPS

Important The CLI commands available for non-CUPS Virtual APN feature is applicable in CUPS environment.

Following are sample configuration for:

1. **Control Plane node**:

```
configure
   context context_name
      apn apn_name
         pdp-type ip_address
         bearer-control-mode mixed
         selection-mode sent-by-ms
         ims-auth-service service_name
            exit
         ip access-group acl_group_name in
         ip access-group acl_group_name out
         authentication pap preference chap preference allow-noauth
         ip context-name context_name
         virtual-apn preference preference apn apn_name
         bearer-access-service  service_name
         cc-profile  cc_profile_index
         [ pdp-type { ipv4 | ipv6 | ipv4v6 } ]
         [ roaming-mode { home | roaming | visiting } ]
         [ serv-gw-plmnid mccmcc_number mnc mnc_number ]
         end
```

Note **bearer-access-service** *service_name*: Specifies the Bearer Access Service (GGSN/P-GW/Other) name. This service name is unique across the context. *service_name* must be an alphanumeric string of 1 through 63 characters.

cc-profile *cc_profile_index*: Specifies the charging characteristics (CC)-profile index. *cc_profile_index* must be an integer from 1 to 15.

[pdp-type { ipv4 | ipv6 | ipv4v6 }]: Configures pdp-type rule. The available options include:

- **ipv4**: Configures VAPN Rule for IPv4.
- **ipv4v6**: Configures VAPN Rule for ipv4v6.
- **ipv6**: Configures VAPN Rule for IPv6.

[roaming-mode { home | roaming | visiting }]: Supports separate PDP context or PDN connection processing for roaming, visiting, and home subscribers.

serv-gw-plmnid: Specifies the Serving Gateway PLMN ID.

```
configure
  context context_name
    apn apn_name
      pdp-type ipv4 ipv6
      bearer-control-mode mixed
      selection-mode sent-by-ms
      ims-auth-service service_name
        exit
      ip access-group acl_group_name in
      ip access-group acl_group_name out
      authentication pap preference chap preference allow-noauth
      ip context-name context_name
      end
```

- **For Gx-based Virtual APN selection**:

```
configure
  context context_name
    ims-auth-service service_name
      policy-control
      diameter encode-supported-features virtual-apn
      end
```

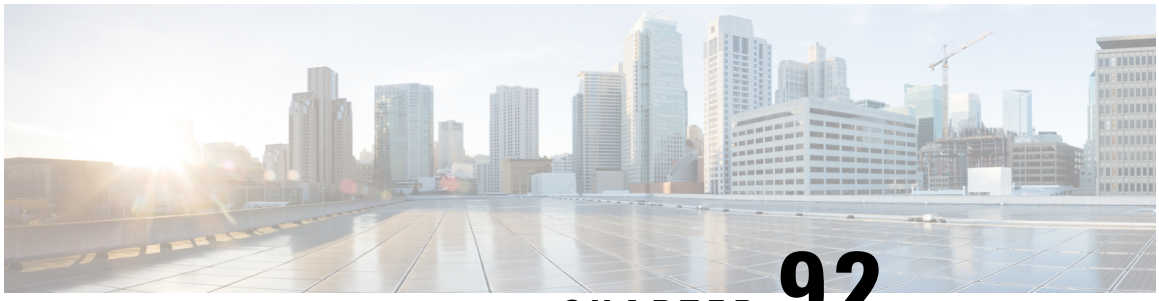
- **For Location-based Virtual APN Selection for GGSN Calls**:

```
configure
  context context_name
    apn apn_name
      virtual-apn preference priority apn vapn_name
routing-area-code-range from start_value to end_value
      end
```

2. **User Plane node**:

```
configure
   context context_name
      apn apn_name
         ip context-name context_name
         end
```

```
configure
   context context_name
      apn apn_name
         ip context-name context_name
         end
```



CHAPTER 92

VoLTE Support in CUPS

- Revision History, on page 787
- Feature Description, on page 787
- How It Works, on page 788
- Limitations, on page 790

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

VoLTE is now supported for P-GW (Pure-P) and SAE-GW (Collapsed) calls in the UPC CUPS Architecture.

With this release, the following functionalities are supported in this feature:

- SRVCC/CSFB support for VoLTE
- Support Suspend notification procedure
- Support Resume Notification procedure
- P-CSCF address selection.
- P-CSCF restoration.
- AF-Charging-ID support.
- Intelligent Graceful Shutdown support.
- PDN Reactivation support for IMS PDN

- Non-Standard QCI support

Relationship

This feature is related to *Priority Recovery Support for VoLTE Calls*.

How It Works

The functioning of VoLTE in CUPS is implemented at a minimal level in this release.

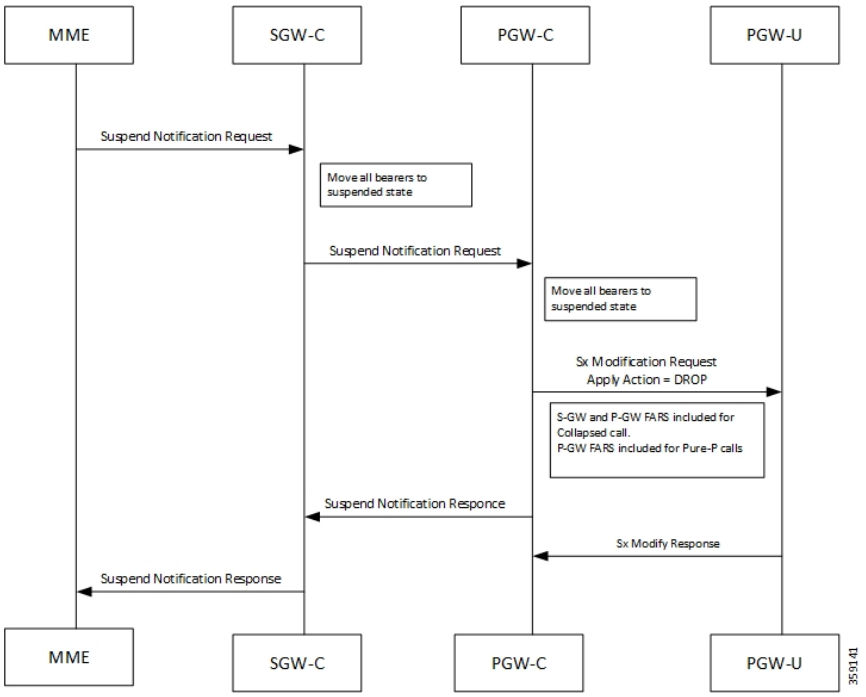
- Suspend Notification for Pure-P and Collapsed calls
- Resume Notification for Pure-P and Collapsed calls

Call Flows VoLTE Support

The following section illustrates call flows that are in support of the VoLTE feature.

Handling Suspend Notifications

The following call flow illustrates Suspend Notifications for Pure-P and Collapsed calls.



On receiving a Suspend Notification message, the PGW-C requests the PGW-U to discard packets received for the suspended PDN connection by setting the DROP flag in the Apply Action IE of the FARs of the corresponding PFCP session.

As part of the suspend notification, the following actions are sent for uplink and downlink data:

- S-GW uplink FARS - Forward Action

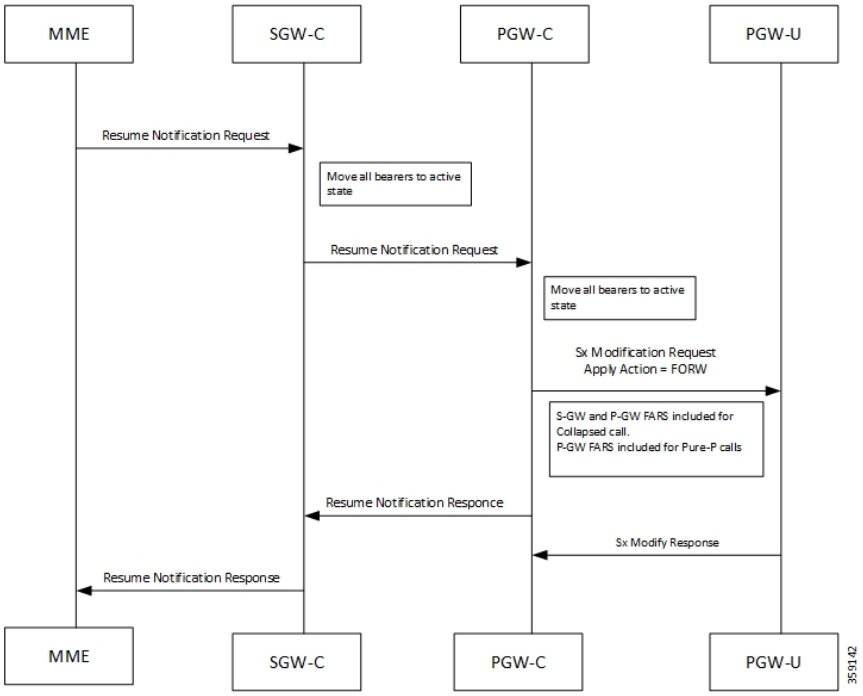
- S-GW downlink FARS - Drop Action
- P-GW uplink FARS - Drop Action
- P-GW downlink FARS - Drop Action

The following conditions are also implemented:

- If SGW receives ULI/RAT/TZ Reporting MBR in Suspended state, all bearers are moved in to active state and forwards MBR to PGW.
- If PGW receives ULI/RAT/TZ Reporting MBR in Suspended state, all bearers are moved in to active state.
- On Receiving suspend notification Session idle timeout is stopped. If PGW receives Empty MBR in Suspended state, all bearers are moved in to active state.

Handling Resume Notifications

The following call flow illustrates Resume Notifications for Pure-P and Collapsed calls.



On receiving the request to resume the PDN connection, the PGW-C re-allows the PGW-U to forward the packets for the PDN connection by:

- setting the FORW flag in the Apply Action IE of the FARs of the corresponding PFCP session or
- setting the gate fields in the Gate Status IE of QERs to the value OPEN.

As part of the resume notification, the following actions are sent for uplink and downlink data:

- P-GW uplink FARS - Forward Action
- P-GW downlink FARS – Forward Action

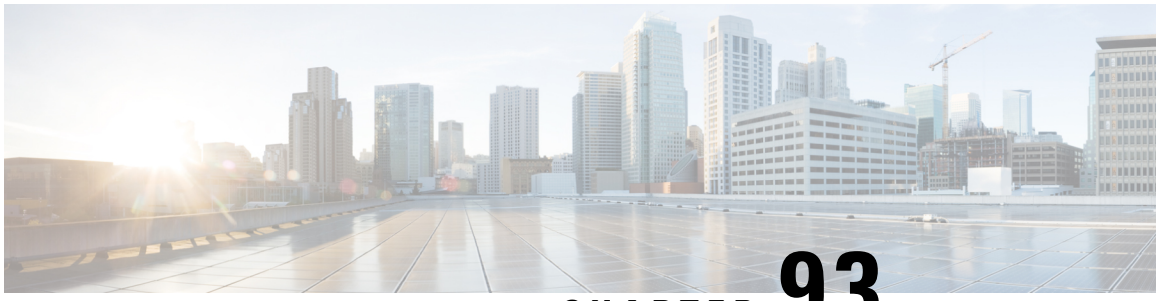
- S-GW uplink FARS - Forward Action
- S-GW downlink FARS – Forward Action

Note On receiving Resume notifications, Session Idle timeout is restarted.

Limitations

The VoLTE support in CUPS has the following limitations:

- VoLTE Call Identification support.
- Session Recovery enhancement for VoLTE.
- VoLTE statistics
- Multimedia Priority Service support.



CHAPTER 93

Volume Reporting over Gx

- Revision History, on page 791
- Feature Description, on page 791
- How it Works, on page 792
- Configuring VoGx Monitoring Key Range, on page 793
- Monitoring and Troubleshooting VoGx, on page 794

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
Trigger-based usage reporting for 3G is supported in CUPS.	21.25
First introduced	Pre 21.24

Feature Description

The Volume Reporting over Gx (VoGx) feature provides PCRF the capability to make real-time decisions based on the data usage by subscribers.

This feature is implemented using the existing non-CUPS architecture, for Control Plane. This implementation is done by mapping the existing VoGx framework and the CUPS data structures such as FAR, PDR, URR and so on.

Important Volume Reporting over Gx is applicable only for volume quota.

How it Works

The following steps explain how Volume Reporting over Gx works:

1. PCEF, after receiving the message from PCRF, parses the usage monitoring-related AVPs and sends the information to IMSA.
2. IMSA updates the information to ECS.
3. After the ECS is updated with the usage monitoring information from PCRF, the PCEF (ECS) starts tracking the data usage.
4. For session-level monitoring, the ECS maintains the amount of data usage.
5. For PCC rule monitoring, usage is monitored with the monitoring key as the unique identifier. Each node maintains the usage information per monitoring key. When the data traffic is passed, the usage is checked and reported against the usage threshold values.

Note In releases earlier than 21.22, the monitoring key value was in the range of 0-134217727.

In 21.22 and later releases, the monitoring key value is in the range of 1-4000000000.

6. The PCEF continues to track data usage after the threshold is reached and before a new threshold is provided by the PCRF. If a new usage threshold is not provided by the PCRF in the acknowledgment of an IP-CAN Session modification where its usage was reported, then usage monitoring does not continue in the PCEF for that IP CAN session.

For additional information about this feature, refer the *SAEGW Administration Guide*.

Supported Standards

The Volume Reporting over Gx feature is based on the following standard: 3GPP TS 29.212 V9.5.0 (2010-06): 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Policy and Charging Control over Gx reference point (Release 9).

Control Plane Handling for VoGx

URR Creation during Session Setup

- Sx Session establishment request is used as per the GxSPI framework.
- The Control Plane function sends the list of URRs in the Sx Session Establishment request, along with their references in corresponding PDRs.

URR Processing in Detach Request

- The URR information will be sent by PGW-U as part of Sx Session Delete Response.
- PGW-C maps these URRs to the corresponding Monitoring-key Buckets and sends the CCR-T containing Usage Report.

Sx Session Report Request

PGW-U sends the Usage Report for Volume Threshold. PGW-C maps the URRs to corresponding Monitoring-key Buckets and generate the Gx CCR-U accordingly.

User Plane Handling for VoGx

Volume Threshold Breach

When data packets match a particular PDR and the PDR has associated URRs that have the measurement method set as Volume, the uplink and downlink usage counters are incremented depending on the PDR source interface type. Once a volume threshold is breached for a particular URR, an Sx Session Report Request message is generated and sent with Usage Report Trigger set as Volume Threshold. All the usage counters of the URRs that are reported is cleared once the message is generated and sent to Control Plane. However, the existing threshold limit will be applicable for further transactions.

Limitations

The VoGx feature has the following limitations.

- Reporting of usage to PCRF during following event triggers are not supported in CUPS:
  - Trigger
    - PGW_TRACE_CONTROL (24)
    - QOS_CHANGE_EXCEEDING_AUTHORIZATION (11)
    - APN_AMBR_MODIFICATION_FAILURE (29)
    - CHARGING_CORRELATION_EXCHANGE (28)
    - OUT_OF_CREDIT (15)
    - REALLOCATION_OF_CREDIT (16)
    - UE_IP_ADDRESS_ALLOCATE (18)
    - UE_IP_ADDRESS_RELEASE (19)
    - APPLICATION_START (39)
    - APPLICATION_STOP (40)
    - REVALIDATION_TIMEOUT (17)
- Trigger-based usage reporting is not supported for 3G in CUPS.

Configuring VoGx Monitoring Key Range

From Release 21.22 onwards, it is mandatory to define the **monitoring-key urr-id-prefix** entries for all the monitoring keys configured locally in the PCEF as part static and predefined rules.

Use the following configurations to enable the monitoring key range.

```
configure
   active-charging service service_name
      mon-key-urr-list list_name
         monitoring-key value urr-id-prefix urr_id
         end
```

NOTES:

- **mon-key-urr-list** *list_name*: Specifies the name of monitoring key list. *list_name* must be a string of size 1-63.
- **monitoring-key** *value*: *value* must be an integer in the range of 1-4000000000.
- **urr-id-prefix** *urr_id*: *urr_id* must be an integer in the range of 1-8388607.
- Multiple monitoring key and URR ID combinations under the list name can be configured. The recommended limit is 2500 entries.
- This CLI command can be configured in both Control Plane and User Plane. After configuring the CLI command in Control Plane, it is mandatory to push the configuration to User Plane using PFD push mechanism. For RCM, it is mandatory to configure **require rcm-configmgr** on User Plane before configuring the CLIs. Both Control Plane and User Plane must be configured through CLI in RCM configuration.
- You should configure only unique monitoring key and URR-ID combinations. These URR-IDs configured through **mon-key-urr-list** should not coincide with the URR-IDs configured through **urr-list**. If such a configuration is attempted, the CLI throws an error.
- If there is a run-time addition of this CLI at Control Plane, it is necessary to push the CLIs using PFD push mechanism so that configurations can be updated at both ends. These configurations will apply next call onwards or at the time of next URR creation.

Monitoring and Troubleshooting VoGx

This section provides information about the CLI commands available for monitoring and troubleshooting VoGx in CUPS.

Show Commands and/or Outputs

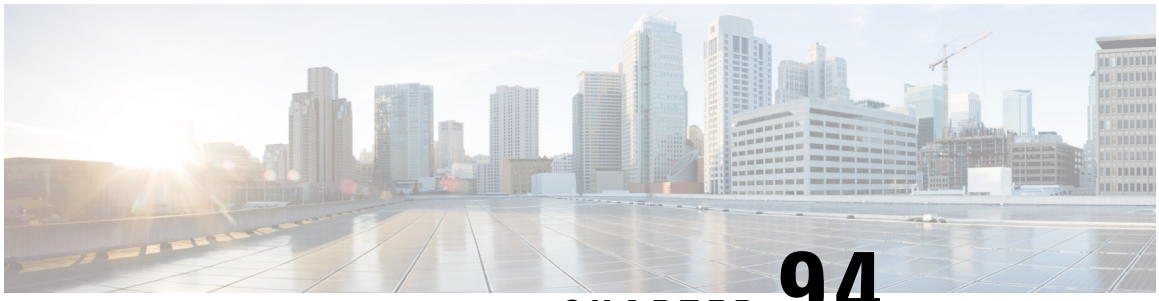
show active-charging subsystem all debug-only

The output of this CLI command has been enhanced to include the following fields in support of VoGx feature in CUPS.

- Total Mon-Key Urr Entries in list
- Total Mon-Key lookup success
- Total Mon-Key lookup failure

show user-plane-service monitoring-key-urr-id-list all

Use this CLI command to view all the monitoring keys that were pushed from Control Plane to User Plane.



CHAPTER 94

VPN Manager Recovery Support

- Feature Summary and Revision History, on page 797
- Feature Description, on page 797

Feature Summary and Revision History

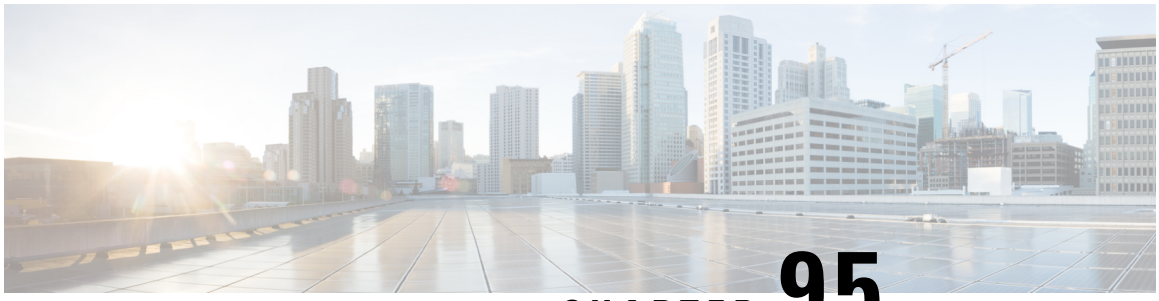
Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The VPN Manager Recovery Support feature enables the recovery of chunks after a VPN Manager (vpnmgr) crash. To recover chunks after the crash, the chunks are stored and allocated to a particular VPN Manager in the local VPN Manager.

When a pool VPN Manager crashes, it recovers the chunk from the local VPN Manager and all the used IPs from all the Session Managers.



CHAPTER 95

VPP Support

Vector Packet Processing (VPPMOB) is a mobility-centric solution based on fd.io's VPP, an open source solution. It leverages fd.io development, particularly in the areas of IP forwarding, routing, and protocols.

- Revision History, on page 799
- Charging Support, on page 800
- Delay-Charging Via Rule Base, on page 800
- Flow Idle-time Out, on page 801
- HTTP Support, on page 801
- IP Readdressing, on page 801
- DNS Readdress Server List, on page 801
- LTE Handover, on page 803
- Next Hop, on page 803
- PDN Update, on page 803
- Policing, on page 803
- Pure-S Support, on page 804
- Response-based Charging via Service Schema, on page 805
- Response-based TRM via Service Schema, on page 805
- ToS Marking, on page 805
- Volume-based Offload, on page 805
- Supported Functionality, on page 805
- Limitations, on page 806
- Enabling Fast Path in User Plane Service, on page 807
- Enabling VPP on SI Platform, on page 807
- Monitoring and Troubleshooting VPP Fast Path, on page 807
- Support for VPP Configuration Parameters Override, on page 808

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
With this release, support has been added for DNS Readdress Server List.	21.25.4
First introduced	Pre 21.24

Charging Support

Usage Reports are notified to the billing server on call deletion or volume/time threshold breach.

When a stream is created on the User Plane, flows – that involve Charging, are associated with charging-specific operations that are set during the stream-creation. The charging counters for all flows – both offloaded and non-offloaded, are maintained on the Fast Path.

During an overflow in the volume threshold, the Fast Path sends a notification with bucket counters (PUSH mode) and in the case of time threshold hit, Applications reads charging counters from Fast Path (PULL mode). The User Plane aggregates these counters with its respective URRs and triggers usage reports over the Sx interface.

Important In this release, the URR support is there for both Volume and Time Threshold. Multiple SDF and one bearer level URRs are supported.

Delay-Charging Via Rule Base

The flavors of delay-charging supported are as follows:

- Charge-to-application all-packets – All control packets (Handshake, midsession, and tear-down) on flow are charged to the application packet matched charging-action.
- Charge-to-application initial-packets – Handshake packets on flow are charged to the application packet matched charging-action.
- Charge-to-application tear-down-packets – Tear-down packets on flow are charged to the application packet matched charging-action.
- Charge-separate-from-application – All control packets are rule-matched and charged to the highest priority rule.

In all the preceding scenarios only the charging is delayed, but the rule-matching occurs on the packet contents.

Important

- Charge-separate-from-application mid session packets are not supported. For offloaded flow, they continue to match the last matched rule.

When you enable the delay-charging feature, the TCP handshake packet hits the rule when it arrives. The TCP handshake packet hits the IP or TCP rule that is based on the configuration. The **show active-charging** CLI command still sees the TCP handshake packet hitting the default rule. This rule is not considered for

charging until the first L7 packet arrives. Once the first L7 packet hits the L7 rule, while sending the quota request, the L7 packet and the TCP handshake packet get included in the same L7 RG.

Flow Idle-time Out

Configurable idle-time out is supported for the maximum duration of 24 hours. In earlier releases, support was available only for specific set of values.

HTTP Support

Analysis of HTTP traffic and policy matching of such HTTP-based rules is supported in this release. Offloading for HTTP flows is supported only for WebSocket, CONNECT method, or if content is present in request/response.

IP Readdressing

IP readdressing for IPv4 and IPv6 is supported in this release.

IP readdressing is configurable using the charging-rule or post-processing rule associated with the charging-action.

Streams are created on Fast Path for flows that match these rules along with the IP Readdressing operation set. All these flows - both offloaded and non-offloaded – will have IPv4/IPv6 address set in the Fast path.

DNS Readdress Server List

Whenever you use an unauthorized DNS server, the request is modified to readdress the DNS IPs to use the authorized servers. **Ruledef** determines if a packet belongs to a DNS query and if the DNS query belongs to a set of authorized DNS servers or not. If the DNS query does not belong to the authorized DNS servers, the flow action is to pick up DNS servers from the **readdress-server-list**.

A **readdress-server-list** is configured under the active charging server. When the flow matches a **ruledef**, the flow action can be configured to use the servers from the **readdress-server-list**.

Configure the **readdress-server-list** under **active-charging service** as follows:

```
configure
  active-charging service service_name
    readdress-server-list name_of_list
      server ipv4_address [ port ]
      server ipv6_address [ port ]
```

Note A maximum of 10 servers can be configured in a **readdress-server-list** and a maximum of 10 **readdress-server-lists** can be configured under active-charging service. Both IPv4 and IPv6 addresses can be configured in the same **readdress-server-list**.

Select the **readdress-server-list** from the list using one of the following two ways:

- Round-robin—Server selection occurs in a round-robin manner for every new flow. Inactive servers in the list are not considered during the selection.
- Hierarchy—The servers that are tagged in this approach are primary, secondary, tertiary, and so on, depending on the order they are defined in the **readdress-server-list**. All flows are readdressed to the primary server as long as it is available. If the Primary server goes down, then flows are readdressed to the secondary server and the same logic recurs. Once, the primary server is active then flows switch back to the primary server for readdressing.

The following CLI command defines the approach for a server selection.

```
charging-action action_name
   flow action readdress-server-list name_of_list [ hierarchy | round-robin
 ]
```

The **round-robin** option is considered as the default option, when no option is provided in the CLI command that is mentioned in the preceding code.

Configure the following CLI command under active-charging service.

```
configure
   active-charging service service_name
      readdress-server-list name_of_list
         server ipv4_address [ port ]
         server ipv6_address [ port ]
         consecutive-failures integer_value
         response-timeout integer_value
         reactivation-time integer_value

         charging-action action_name
            flow action readdress server-list name_of_list
      exit
```

Note Consider the following values to configure the CLI command mentioned in the preceding code.

- **consecutive-failures**—Integer value must range between 1–10. The default value is *5*.
- **response-timeout**—Integer value must range between 1–10000 milliseconds. The default value is *1000*.
- **reactivation-time**—Integer value must range between 1–1800 seconds. The default value is *300*.

Readdress Server States

The readdress server states are described as follows:

- Active state—Once configured, all servers are marked as Active.
- Inactive state—If no response is received from the readdressed server, then the server is marked as Inactive.

- Active-Pending state—Once the server is in Active-Pending state, it is available to accept the requests for readdressing. In this state, if a request is readdressed to this server and response is returned from it, then the server-state is changed to Active. Otherwise, it is moved back to Inactive state.

LTE Handover

The following types of handovers are supported:

- S-GW Relocation for X2 based handovers (OI set to 1).
- S-GW Relocation for S1 based handovers (OI set to 0).
- eNodeB F-TEIDu Update.

For S-GW relocation, the following combinations are supported:

- P-GW anchored call.
- P-GW anchored call to Collapsed call.
- Collapsed call to P-GW anchored call.

Next Hop

Next hop address for IPv4 and IPv6 is supported in this release.

The Next-Hop address is configurable using the charging-rule or post-processing rule associated with the charging-action.

Streams are created on Fast Path for flows that match these rules along with the Next Hop operation set. All these flows - both offloaded and non-offloaded – will have Next Hop address set in the Fast path.

PDN Update

PDN Update procedures are supported with VPP in this release.

All flows are onloaded to SM-U whenever Rule Addition/Modify/Removal is received through any Gx procedures. All the packets on these onloaded flows are then sent to SM-U. The flows are also onloaded when transport level marking and charging parameters changes for the flow. These flows are again offloaded on the packet for which rule-match changes, or Transaction Rule Matching (TRM) engages again.

Policing

The policer configuration uses inputs from the session manager, these inputs are received either from PCRF as AMBR or from flow-level QoS information. The values received from the PCRF is always accepted for session level AMBR policing. But, the flow-level policing is prioritized, if available, and sequentially the AMBR policing is applied. In other words, the policer engine applies the hierarchical policing - first the flow-level/rule bandwidth limiting and then the session level bandwidth limiting.

Note AMBR modifications during session run-time through RAR or CCA-U is applicable.

The input values received from the session manager are pushed into a policer configuration and a policer token bucket. For each direction - uplink or downlink, a new record is created for Policer configuration and Policer token bucket.

The Policer configuration is the reference for the policer engine, and the policer token bucket is used for calculation and restoration of values.

Currently, Policing is supported for AMBR received from PCRF and Rule-level QoS information for dynamic rules. For static and predefined rules, bandwidth limiting is achieved by the bandwidth policy configuration. Extended bit rates configured in bandwidth-policy configuration in Active Charging Service Configuration mode on Control Plane is provided to the User Plane as part of the configuration push mechanism, and same is applied for policing by User Plane. The following is an example configuration of bandwidth policy:

```
configure
   active-charging service ACS
      bandwidth-policy BWP

         flow limit-for-bandwidth id 1 group-id 1

         flow limit-for-bandwidth id 2 group-id 2
         group-id 1 direction uplink peak-data-rate 256000 peak-burst-size 32000
violate-action discard
         group-id 1 direction downlink peak-data-rate 256000 peak-burst-size 32000
violate-action discard
         group-id 2 direction uplink peak-data-rate 128000 peak-burst-size 16000
violate-action discard
         group-id 2 direction downlink peak-data-rate 56000 peak-burst-size 7000
violate-action discard
         exit
```

Limitations

In this release, Policing has the following limitations:

- Modification of **bandwidth-policy** is not supported.
- Interaction with other features such as - ITC bandwidth limiting, token replenishment (both APN level and ACL level) is not supported.
- Currently, policer-based statistics are not supported.

Note As policer statistics are not yet supported, the operator can verify bandwidth limiting using network performance monitoring tools.

Pure-S Support

Pure-S default bearer VPP integration is now supported in the CUPS Architecture. Earlier, Pure-S calls on CUPS were supported using IFTASK. Now, Pure-S call data path also uses VPP.

As part of VPP integration for Pure-S calls, calls on SAEGW-UP will install one bearer stream (3 Tuple – GTPU Service IP address, TEID, VRF id) per direction and also one TEP row per direction is created.

Supported Functionality:

Supported functionality for Pure-S includes:

- Most procedures for Collision between MME and Network Initiated scenarios (MBR/CBR/UBR/DBR).
- DBCmd and BRCmd commands.
- SAEGW-UP supports movement of IP transport from IPv4 to IPv6, or IPv6 to IPv4 during IDLE to ACTIVE transition, and handover procedures on S1-u interface. Transport selected on S1-u at the time of attach is also supported. For example, eNode handover from IPv4 eNodeB to IPv6 eNodeB.

Response-based Charging via Service Schema

HTTP Request is charged to the HTTP Response matched charging-action.

Response-based TRM via Service Schema

The Transaction Rule Matching (TRM) on uplink stream is engaged only after the HTTP response is received.

ToS Marking

Feature Description

ToS Marking for IPv4 and IPv6 is supported in this release.

The inner IP ToS marking address is configurable using the charging-rule or post-processing rule associated with the charging-action. The outer IP ToS marking is performed using the QCI-DSCP marking table configured on the control plane.

Streams are created on Fast Path for flows that match these rules along with the operations set. All these flows - both offloaded and non-offloaded – will have IPv4/IPv6 ToS marking set in the Fast path.

Volume-based Offload

In case of HTTP protocol, the content in request/response (if present) gets offloaded to fastpath for each transaction in a flow. The last packet of the content switches back the stream to passive state and the packet reaches the Session Manager.

Supported Functionality

The following call flavors are supported in this release:

- Pure-P IPv4/IPv6 calls.

- Collapsed IPv4/IPv6 calls.
- Default bearer.
- Pure-S functionality.
- Dedicated bearer.
- Handovers.

The following functionalities are supported in this release:

- ToS marking of the payload packets (Charging action) and outer GTP-U packets (QCI/QoS mapping table).
- Next hop feature (IPv4/IPv6).
- IP Readdressing feature (IPv4/IPv6).
- Post processing rules with action as discard.
- Post Processing rules with action as Next hop forwarding (IPv4/IPv6).
- Post Processing rules with action as ToS marking (UL, and DL).
- Post Processing rules with action as Readdressing (IPv4/IPv6).
- URR functionality (Gz only) - One SDF, and one bearer level URR.
- Only Gz charging is supported.
- Fragmentation and reassembly is supported in VPP.
- HTTP traffic policy match is supported. HTTP offload support is only for CONNECT and WebSocket requests.
- This release has been validated to support up to 5000 flows across all applications per subscriber. Although this limit is not imposed by the software, it is the recommended operating limit. Exceeding this limit may lead to application failures and so, it is recommended that the following CLI be configured in the Rulebase Configuration mode: **flow limit-across-applications** *5000*.

Limitations

The following functionalities are not supported in this release:

- Gy and Rf are supported independently, however, they both cannot be enabled at the same time for the same subscriber.
- Fast Path CLI can be disabled if it was previously enabled. However, User Plane must be reloaded.
- **VPP crashlog support**: Generation of crash records and mini-core files are supported. Generation of full core files for VPP is not supported.

Enabling Fast Path in User Plane Service

Use the following CLI commands to enable Fast Path (VPP) in User Plane service.

```
configure
  context context_name
    user-plane-service service_name
      associate fast-path service
      end
```

NOTES:

- **fast-path**: Specifies the Fast Path related parameters.
- **service**: Specifies the Fast Path related configurations.

Enabling VPP on SI Platform

To launch VPP:

1. Log on to host machine, and create an ISO image that contains the file: *staros_param.cfg*
2. Create a file that has the line: FORWARDER_TYPE=vpp
3. Create an ISO file containing the *staros_param.cfg* file:

   ```
   genisoimage -l -o ssi_vpp.iso -r vppiso/
   ```

   If genisoimage is not installed, execute:

   ```
   sudo apt-get install genisoimage
   ```

4. Stop the VM if it is running:

   ```
   virsh destroy <vm_name>
   ```

5. If a disk is already attached to the VM that does not have VPP identified as the forwarder, then detach the disk.

   Run the **dumpxml** command on the VM to see if there is a disk attached.

   To detach the disk, execute:

   ```
   virsh detach-disk <vm_name> hdc –config
   ```

6. Attach the ISO file that contains the *staros_param.cfg* file:

   ```
   virsh attach-disk <vm_name> <Path_of_ISO_FILE> hdc –type cdrom –config
   ```

Monitoring and Troubleshooting VPP Fast Path

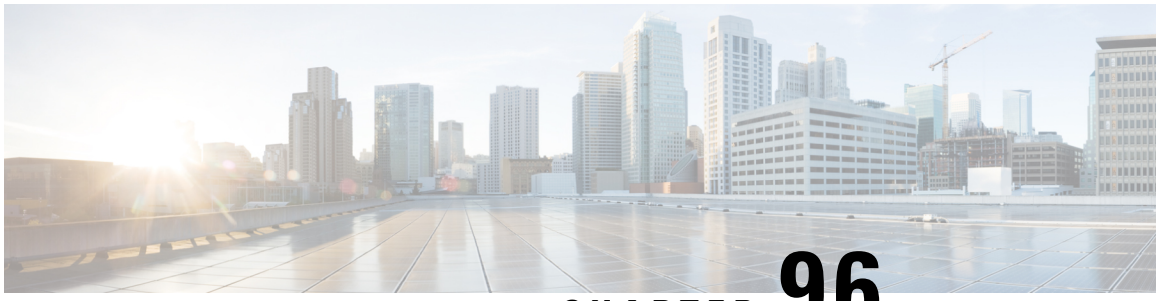
To determine if the flows are offloaded, check for Fast Path statistics in the output of the following CLI commands:

- **show subscribers user-plane-only full all**

- **show user-plane-service all**
- **show user-plane-service statistics analyzer name ip**
- **show user-plane-service statistics analyzer name ipv6**
- **show user-plane-service statistics analyzer name tcp**
- **show user-plane-service statistics analyzer name udp**
- **show user-plane-service statistics analyzer name http**

Support for VPP Configuration Parameters Override

To configure the VPP Configuration parameters, see the *VPC-SI Administration Guide*. These parameters can be overriden. Ensure that you contact your Cisco account representative to assist in identifying the override values.



CHAPTER 96

VRF Support for CUPS

- Revision History, on page 809
- Feature Description, on page 809
- Configuring VRF, on page 811
- Monitoring and Troubleshooting, on page 813

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

The VRF Support for CUPS feature enables association of IP pools with virtual routing and forwarding (VRF). These IP pools are chunked like any pools. The chunks from this pool are allocated to the User Planes (UPs) that are configured to use these pools. As in the existing deployment, VRF-associated pools in CUPS can only be of type—STATIC or PRIVATE.

The chunks from the PRIVATE VRF pool are allocated when the UP comes for registration similar to the normal private pools. The chunks from the STATIC VRF pool are allocated only when calls come up for that chunk, similar to normal static pools.

Note VRF limit per UP is 205.

Overlapping Pools in Same UP

Overlapping pools share and use an IP range. Overlapping pools can either be of type STATIC or PRIVATE. No public pools can be configured as overlapping pools. Each overlapping pool is part of different VRF

(routing domain) and pool-group. Since an APN can use only one pool-group, overlapping pools are part of different APN as well.

Without this functionality, overlapping pools can be configured at CP but chunks from two overlapping pools can't be sent to same UP. That is, the UP can't handle chunks from two different overlapping pools. So, same number of UPs and overlapping pools are required for sharing same IP range.

With this functionality, UP can handle chunks from two different overlapping pools. So, a single UP can handle any number of overlapping pools sharing the same IP range.

Note Only VRF-based overlapping pools are supported in CUPS. Other flavors of overlapping pools, like NH-based, VLAN-based, and so on, aren't supported in CUPS.

The functionality of overlapping pools in same UP includes:

- When a chunk from particular pool is installed on an UP, its corresponding vrf-name is sent along with the chunk.
- The UPs are made VRF-aware of chunks and therefore, UPs install chunks on the corresponding VRFs and the chunk database is populated under the VRFs.
- During call allocation, release, recovery, or any communication towards VPNMgr, the corresponding SessMgr at UP includes vrf-id. This enables VPNMgr to pick the correct chunk for that IP under the provided vrf-id for processing.

VPNMgr Crash Outage Improvement for IP Pool under VRF

In case of Demux card migration or if VPNMgr goes down, new calls are rejected until VPNMgr rebuilds its database. For enterprise solutions where there are lots of VRFs, the new call impact may be higher than expected.

The Delayed VRF Programming, a CLI-controlled feature, is introduced to reduce the new call impact by delaying the programming of IP pool VRFs during VPNMgr recovery (restart and switchover) scenarios.

Configuring Delayed VRF Programming

Use the following CLI commands to enable faster recovery of VPNmgr with VRF with IP pool configured on it in CP and UP.

```
configure
  context context_name
    ip vrf vrf_name
      ip delay-vrf-programming-during-recovery
      end
```

NOTES:

- By default, the keyword/feature is disabled.
- The CLI keyword is applicable to both CP and UP VRF configurations.
- Enabling the feature on non-IP pool VRFs isn’t recommended.

- It's assumed that the IP pool VRF won't have any other control protocols (such as SRP) enabled, which requires TCP connections/kernel interactions.
- During the delayed interval:
  - Any functionality which requires kernel interaction for recovering VRF will not work. No subscriber data outage is expected.
  - Any configuration change related to Route/BGP/BFD/Interface/VRF fails and configuration must be reapplied.

Change in CLI Syntax

As part of this feature, the syntax of **show ip vrf** *vrf_name_string* CLI command is changed for all platforms, including non-CUPS.

Following is the new syntax: **show ip vrf name** *vrf_name_string*

Also, all existing optional keyword after **show ip vrf** *vrf_name_string* is changed to **show ip vrf name** *vrf_name_string*. However, there's no change in output of the CLI commands.

Configuring VRF

Follow these steps to implement VRF support for CUPS.

At Control Plane:

1. Associate the IP pool with VRF.
2. Create an APN to use this pool.
3. Associate UP with UP Group to ensure that the UP uses only the specific APN.

   If there are overlapping pools, ensure that you create separate APNs for each one of the pools. Also, ensure that different UPs use each of these APNs.

The following is a sample of the CP configuration:

```
context EPC2
  apn mpls1.com
    pdp-type ipv4 ipv6
    bearer-control-mode mixed
    selection-mode subscribed sent-by-ms chosen-by-sgsn
    ims-auth-service iasGx
    ip access-group css in
    ip access-group css out
    ip context-name isp
    ip address pool name PRIVATE
    ipv6 address prefix-pool PRIVATEV6
    ipv6 access-group css6 in
    ipv6 access-group css6 out
    cc-profile any prepaid-prohibited
    active-charging rulebase cisco
    user-plane-group mpls1
  exit
  apn mpls2.com
    pdp-type ipv4 ipv6
    bearer-control-mode mixed
```

```
      selection-mode subscribed sent-by-ms chosen-by-sgsn
      ims-auth-service iasGx
      ip access-group css in
      ip access-group css out
      ip context-name isp
      ip address pool name PRIVATE_1
      ipv6 address prefix-pool PRIVATEV6_1
      ipv6 access-group css6 in
      ipv6 access-group css6 out
      cc-profile any prepaid-prohibited
      active-charging rulebase cisco
      user-plane-group mpls2
    exit

config
  context isp
    ip vrf mpls-vrf-1
    ip vrf mpls-vrf-2
    #exit

    #exit
    cups enable
    ip pool PRIVATE 209.165.200.225 255.255.255.224 private 0 chunk-size 64 vrf mpls-vrf-1
   ip pool PRIVATE_1 209.165.200.225 255.255.255.224 private 0 chunk-size 64 vrf mpls-vrf-2

    ip pool STATIC 209.165.200.226 255.255.255.224 static vrf mpls-vrf-1
    ipv6 pool PRIVATEV6 prefix 8001::aaaa/54 private 0 chunk-size 64 vrf mpls-vrf-1
    ipv6 pool PRIVATEV6_1 prefix 8001::aaaa/54 private 0 chunk-size 64 vrf mpls-vrf-2
    ipv6 pool v6pool2 prefix 2a02:2121:2c4::/46 static 0 vrf mpls-vrf-1
exit

  user-plane-group mpls1
    peer-node-id ipv4-address 209.165.200.226
  #exit
  user-plane-group mpls2
    peer-node-id ipv4-address 209.165.200.228
  #exit
```

At User Plane:

It's recommended to configure VRF in UP before chunk is pushed from CP. Else, it leads to the failure of complete IP pool transaction (including chunks that don't belong to the VRF), and retry attempt by CP after some time.

The following is a sample of the UP configurations:

User-Plane 1:

```
Config
  context EPC2
    sx-service sx
      instance-type userplane
      bind ipv4-address 209.165.200.226 ipv6-address bbbb:aaaa::4
    exit
    user-plane-service up
      associate gtpu-service pgw-gtpu pgw-ingress
      associate gtpu-service sgw-ingress-gtpu sgw-ingress
      associate gtpu-service sgw-engress-gtpu sgw-egress
      associate gtpu-service saegw-sxu cp-tunnel
      associate sx-service sx
      associate fast-path service
      associate control-plane-group g1
    exit

  context isp
```

```
    ip vrf mpls-vrf-1
    #exit
    ip vrf mpls-vrf-2
    #exit
    apn mpls1.com
      pdp-type ipv4 ipv6
      bearer-control-mode mixed
      selection-mode sent-by-ms
      ip context-name isp
    exit
exit
control-plane-group g1
    peer-node-id ipv4-address 209.165.200.227
  #exit
  user-plane-group default
```

User-Plane 2:

```
Config
  context EPC2
    sx-service sx
      instance-type userplane
      bind ipv4-address 209.165.200.228 ipv6-address bbbb:aaaa::5
    exit
    user-plane-service up
      associate gtpu-service pgw-gtpu pgw-ingress
      associate gtpu-service sgw-ingress-gtpu sgw-ingress
      associate gtpu-service sgw-engress-gtpu sgw-egress
      associate gtpu-service saegw-sxu cp-tunnel
      associate sx-service sx
      associate fast-path service
      associate control-plane-group g1
    exit
exit

  context isp
    ip vrf mpls-vrf-1
    #exit
    ip vrf mpls-vrf-2
    #exit
    apn mpls2.com
      pdp-type ipv4 ipv6
      bearer-control-mode mixed
      selection-mode sent-by-ms
      ip context-name isp
    exit
exit

control-plane-group g1
    peer-node-id ipv4-address 209.165.200.228
  #exit
  user-plane-group default
```

Monitoring and Troubleshooting

This section provides information regarding the CLI command available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show ip chunks

The output of this CLI command displays all the chunks in that context.

With Overlapping Pools in Same UP functionality, VRF option is introduced in the CLI, **show ip chunks vrf** *vrf_name*, that displays only the chunks under that VRF.

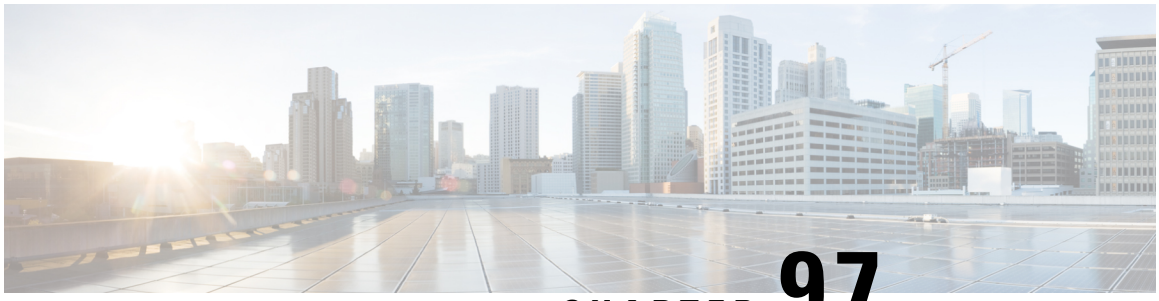
- chunk-id
- chunk-size
- vrf-name
- start-addr
- end-addr
- used-addrs
- Peer Address

show ipv6 chunks

The output of this CLI command displays all the chunks in that context.

With Overlapping Pools in Same UP functionality, VRF option is introduced in the CLI, **show ipv6 chunks vrf** *vrf_name*, that displays only the chunks under that VRF.

- chunk-id
- chunk-size
- vrf-name
- start-prefix
- end-prefix
- used-prefixes
- Peer Address



CHAPTER 97

X-Header Insertion and Encryption

- Revision History, on page 815
- Feature Description, on page 815
- How It Works, on page 815
- Configuring X-Header Insertion and Encryption, on page 816
- Monitoring and Troubleshooting the X-Header Insertion and Encryption feature, on page 819

Revision History

Revision Details	Release
First introduced.	21.25

Feature Description

The X-Header Insertion and X-Header Encryption features is collectively known as Header Enrichment. This feature enables in appending headers to HTTP or WSP GET and POST request packets, and HTTP Response packets for use by end applications, such as mobile advertisement insertion (MSISDN, IMSI, IP address, user-customizable, and so on).

How It Works

X-Header Insertion

This section provides an overview of the X-Header insertion feature.

Extension header (X-Header) fields are fields that are not defined in RFCs or standards but can be added to protocol headers for specific purposes. The X-Header mechanism allows additional entity-header fields to be defined without changing the protocol, but these fields cannot be assumed to be recognizable by the recipient. The unrecognized header fields must be ignored by the recipient and must be forwarded by transparent proxies.

The X-Header insertion feature enables inserting x-headers in HTTP or WSP GET and POST request packets and HTTP response packets. Operators wanting to insert X-headers in HTTP or WSP request and HTTP

response packets, can configure rules for it. The charging-action associated with the rules contain the list of X-headers to be inserted in the packets.

X-Header Encryption

This section provides an overview of the X-Header Encryption feature.

X-Header encryption enhances the X-header insertion feature to increase the number of fields that can be inserted, and also enables encrypting the fields before inserting them.

If X-Header insertion has already happened for an IP flow (because of any X-Header format), and if the current charging-action has the first-request-only flag set, X-Header insertion won't happen for that format. If the first-request-only flag is not set in a charging-action, then for that X-Header format, insertion continues happening in other suitable packets of that IP flow.

Changes to X-Header format configuration will not trigger reencryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next reencryption time to those existing calls for which reencryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value won't be available, insertion of that parameter stops.

Note This feature does not support recovery of flows.

Configuring X-Header Insertion and Encryption

This section describes how to configure the X-Header Insertion and Encryption features, collectively known as Header Enrichment.

X-Header Insertion

Table 48: Procedure

Step	Description
1	Creating/configuring a ruledef to identify the HTTP/WSP packets in which the X-Headers must be inserted.
2	Creating/configuring a rulebase and configuring the charging-action, which inserts the X-Header fields into the HTTP/WSP packets.
3	Creating/configuring the X-Header format.
4	Configuring insertion of the X-Header fields based on the message type in the charging action.

X-Header Encryption

Table 49: Procedure

Step	Description
1	X-Header insertion, encryption, and the encryption certificate are configured in the CLI.
2	When the call gets connected, and after each regeneration time, the encryption certificate is used to encrypt the strings.
3	When a packet hits a ruledef that has x-header format configured in its charging-action, X-Header insertion into that packet is done using the given X-Header-format.
4	If X-Header-insertion is to be done for fields which are marked as encrypt, the previously encrypted value is populated for that field accordingly.

Configuring X-Header Insertion

This section describes how to configure the X-Header Insertion feature.

To configure the X-Header Insertion feature:

Table 50: Procedure

Step 1	Create or Configure a ruledef to identify the HTTP packets in which the X-headers must be inserted.
Step 2	Create or Configure a rulebase and configure the charging-action, which inserts the X-header fields into the HTTP packets.
Step 3	Create the X-header format as described in *Creating the X-Header Format*.
Step 4	Configure the X-Header format as described in *Configuring the X-Header Format*.

Creating the X-Header Format

To create an x-header format, use the following configuration:

```
configure
  active-charging service ecs_service_name
    xheader-format xheader_format_name
    end
```

Configuring the X-Header Format

To configure an x-header format, use the following configuration:

```
configure
  active-charging service ecs_service_name
    xheader-format xheader_format_name
      insert xheader_field_name string-constant xheader_field_value | variable
{ bearer { 3gpp { apn | charging-characteristics | charging-id | imei |
```

```
imsi | qos | rat-type | s-mcc-mnc | sgsn-address } | acr | customer-id |
 ggsn-address | mdn | msisdn-no-cc | radius-string |
radius-calling-station-id | session-id | sn-rulebase |
subscriber-ip-address | username } [ encrypt ] | http { host | url } }
       end
```

Configuring X-Header Encryption

This section describes how to configure the X-Header Encryption feature.

Table 51: Procedure

Step 1	Configure X-Header Insertion as described in *Configuring X-Header Insertion*.
Step 2	Create or Configure a rulebase, and configure the encryption certificate to use and the reencryption parameter as described in *Configuring X-Header Encryption*.
Step 3	Configure the encryption certificate to use as described in *Configuring Encryption Certificate*.

Configuring X-Header Encryption

To configure X-Header Encryption, use the following configuration example:

```
configure
  active-charging service ecs_service_name
    rulebase rulebase_name
      xheader-encryption certificate-name certificate_name
      xheader-encryption re-encryption period re-encryption_period
      end
```

NOTES:

- This configuration enables X-Header Encryption for all subscribers using the specified rulebase.
- If the certificate is removed, ECS continues using the copy that it has. The copy is set free once the certificate name is removed from the rulebase.
- Changes to x-header format configuration won't trigger re-encryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next reencryption time to those existing calls for which reencryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value won't be available, insertion of that parameter stops.

Configuring Encryption Certificate

To configure the encryption certificate, use the following configuration:

```
configure
  certificate name certificate_name pem { { data pem_certificate_data private-key
 pem [ encrypted ] data pem_pvt_key } | { url url private-key pem { [
encrypted ] data pem_pvt_key | url url } }
  end
```

Verifying the X-Header Insertion and Encryption Configuration

Enter the following command in the Exec Mode to verify your configuration:

xheader-format *xheader_format_name*

Monitoring and Troubleshooting the X-Header Insertion and Encryption feature

This section provides information on the show commands and/or their outputs available to support this feature.

show active-charging charging-action statistics name

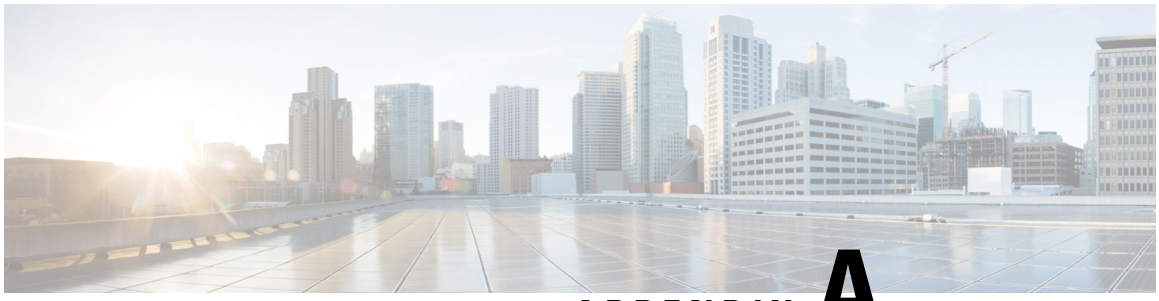
The output of this command displays statistics for X-Header information.

- XHeader Information:
  - XHeader Bytes Injected
  - XHeader Pkts Injected
  - IP Frags consumed by XHeader
  - XHeader Bytes Removed
  - XHeader Pkts Removed

show active-charging rulebase statistics name

The output of this command displays the Header Enrichment statistics.

- HTTP header buffering limit reached



APPENDIX A

IP Pool Planning Guidelines

- IP Distribution in CUPS Architecture, on page 821
- UP Group Concept, on page 821
- When to Add New Pool, on page 822
- IP Pool Fine-tuning Parameters, on page 824
- Dynamic IP Pool Planning Guidelines, on page 825
- Static IP Pool Guidelines, on page 827

IP Distribution in CUPS Architecture

If IP Pool is configured in UP, it becomes tightly coupled to the UP. If the IP pool is unused to a large percentage, it becomes a waste of resources. UPs, which are short of IP resources could benefit from the unused resources, if they are available to them. Similarly, if one UP becomes unreachable, the IP range allocated to it can't be reused. To overcome this limitation, IP chunking mechanism is introduced.

In CUPS architecture, IP pools are configured in the CP, and CP is responsible for the IP Pool management. Any IP pool configured on the CP is divided into chunks of configured size. During the UP-registration process, the CP identifies all the APNs which are being served by that particular UP and the associated IP pool configurations in each APN. The CP allocates the chunks from these IP pools to the UP. CP monitors the chunk usage at per pool level for each UP every **chunk-threshold-timer** second, and if the chunks threshold is reached, it allocates new chunks to the UP. Similarly, if certain IP chunks in the UP aren't utilized, and if CP doesn't have enough free chunks in reserve for that pool, CP withdraws those IP chunk resources from the respective UPs.

In CUPS architecture, IP pools aren't considered as resource/criteria for UP selection. To avoid uneven load distribution across UPs, the operator must ensure that sufficient IP chunks are available for all APNs and UPs in an UP Group.

UP Group Concept

When the UP associates with CP, chunks from IP pool associated with APN are distributed to the UP. UP in turn is associated with single UP Group, which is tied to APN. So, the chunks from IP pool that are configured as part of APN are distributed to the UPs belonging to the UP group associated with that APN. Therefore, the size of UP group must be kept in consideration while planning the pool size and chunk size.

Note UP can only be part of single UP Group. You must avoid adding an UP to two UP groups, because it can cause undefined behavior.

Default UP Group

When an APN doesn't have UP Group configured explicitly, they are considered part of default UP group. A default UP Group is used mostly for consumer traffic where the APNs are too large catering to large number of subscribers and operating on large IP pool.

Specific UP Group

To direct traffic of specific APN to selective set of UPs, specific UP Group is used. For example, consider a case of enterprise subscribers where APN and IP pool are small (order of size 256). In this case, it's better not to chunk the pool and dedicate the entire APN to the UP group consisting of only one UP. Similarly, consider very large deployments consisting of large APNs and 20+ UPs. Even in this case, there's no large APN that it requires all 20 UPs to serve it. To better utilize chunking mechanism of IP pools, divide these UPs and APNs into some specific UP groups.

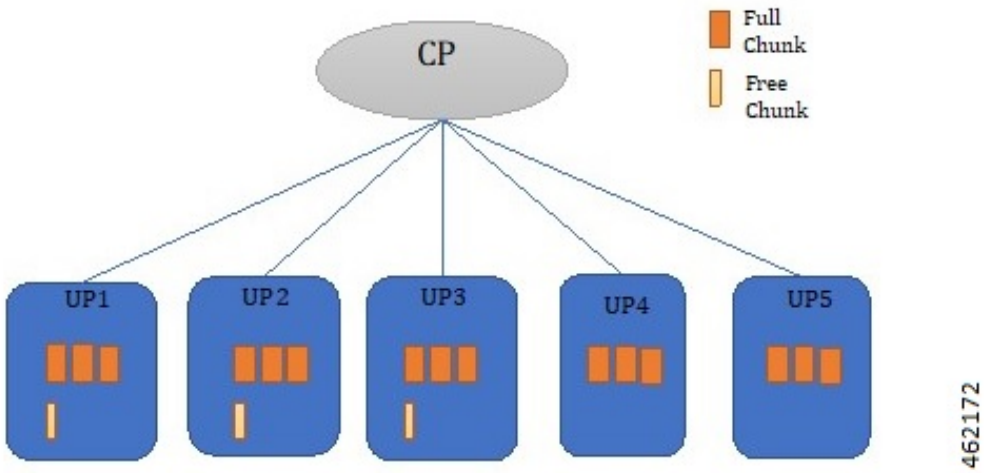
For example, consider a case where you have 20 UPs and require the UPs to serve 100k subscribers each. Consider that there are 25 APNs each having a pool of size 16 K with chunk size of 1k, and six APNs of size 64k, and chunk size of 1k. In this case, to utilize IP pools better, you can create two UP Groups, UP group-1 with four UPs to cater 25 small APNs and UP group-2 consisting of 16 UPs and six large APNs.

When to Add New Pool

When the amount of free chunks left for an APN are less than the number of UPs in the UP Group, and if the selected UP has all chunks fully utilized (that is, no free chunks left), then it leads to uneven load distribution. Also, UP selection algorithms are overridden and IP is given from the UP, which has a chunk available.

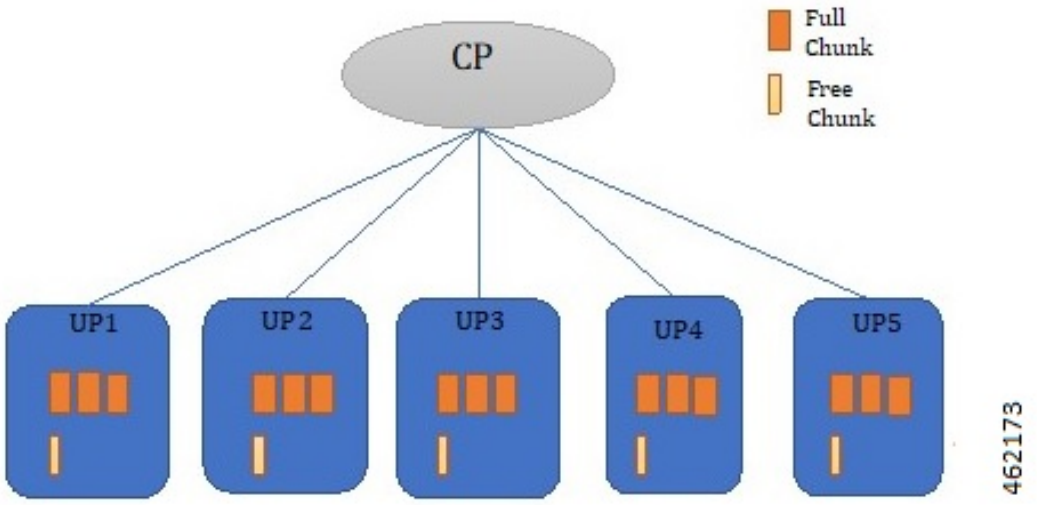
This scenario is illustrated in the following figure:

Figure 44: When to Add New Pool



Add more pools in that APN to avoid uneven load distribution, and chunking should be done to make chunks as even as possible.

Figure 45: Scenario After the New Pool is Added



Note We recommend you to plan the dimensioning in advance for future use, and add more chunks to avoid uneven load distribution.

IP Pool Fine-tuning Parameters

Threshold Timer

CP pushes or removes chunk to/from all UPs periodically. The periodicity is configured with the help of **chunk-threshold-timer**. If the UP has used more than 70% of address allocated from a pool, then a new chunk is pushed to the UP. Similarly, If the UP is underutilizing IP pool resources, then CP pulls back unused chunk from the UP. Removal of chunk is also dependent on other factors described in the section Chunk Withdrawal.

Chunk Withdrawal

CP withdraws chunk from UPs which are underutilizing that pool when CP has free chunks less than **min-chunks-threshold-per-pool** at periodic interval. If the UP is using less than 40 % of allocated IP address from the pool, and if the UP has more than two free chunks available, then a chunk is withdrawn from it.

Initial Chunk Pushed

Initial chunk pushed to each UP can be controlled using **cups max-user-plane** keyword, and it works at context level. Initial chunks are pushed depending on the following factors:

- Total free chunks in pool (command **show ip pool** displays this value)
- Total chunks in pool (value configured in **cups max-user-planes** CLI). By default, the value is set to 10 for max-user-planes.
- Three chunks

Example 1:

The CUPS **max-user-plane** CLI command is used to decide how many initial chunks are pushed to the UP when it registers. The number of initial chunks pushed depends on the following factors:

- If chunks < **max-user-plane** value, then one chunk is pushed.
- If chunks are more, then it's minimum of (chunks/**max-user-plane**, 3).

  Default value of max-user-plane is 10.

Consider that you have eight chunk pools, and want to use four UPs in deployment. You may be interested to give all chunks directly to all UPs. You can do the same by setting **max-user-plane** as 4. Similarly, if you want to have eight UPs in deployment, you can configure the value as 8, and only one chunk is pushed to an UP.

Moreover, it provides some extra chunks which can serve as buffer to cater to sudden surge of subscribers. This surge of subscribers might exceed the normal threshold chunk replenishment rate. Consider that there's one pool with 32 chunks and four UPs of 2k size, and incoming subscribers rate is 2k per minute. If you want to have extra buffer chunk, that is, guard chunk to avoid load imbalance due to incoming subscribers being higher than chunk replenishment rate in small window, then you can use **max-user-plane** CLI. If you don't want to use this CLI, then you can set it to 1000, and always one chunk is pushed.

Chunk Size

Chunk size should be planned considering both CEPS and evenness of chunk with respect to number of UPs in UP Group. Configuring too large chunks can lead to unevenness among UPs with respect to IP addresses. But, if the chunk is too small, then chunk replenishment rate might be lower than incoming subscribers. Low chunk replenishment rate leads to UP override or load imbalance.

Dynamic IP Pool Planning Guidelines

Chunking Guidelines

Chunking should be planned in consideration with the UP-Group scale. There are associated implications of a too large or too small sized chunk, so maintaining a good balance is important.

- Smaller chunk size reduces the imbalance between the chunk distribution among the UPs. But, very small chunk sizes lead to faster chunk exhaustion at the UP, and negatively impacts the CEPS rate.
- Very large chunk size may result in uneven distribution of chunks among the UPs. This issue may lead to UP override or load imbalance on certain UPs where no IPs are available, while IP address is still available with another UP.

The following shows a sample configuration where better IP pool resource planning is recommended:

- **IPv6 pool:** Poolv6_example_1 pool_group_example1 x:x:x:x::/48 chunk_size = 8192
- **IPv6 pool:** Poolv6_example_1 pool_group_example1 x:x:x:x::/48 chunk_size = 8192
- **UPs associated with APN:** UP1, UP2, UP3, UP4, and UP5
- **Threshold timer:** 60 secs
- **Incoming Subscriber rate per UP:** 6k subscribers per minute

(Considering that only one IP pool group is attached to the APN, which this UP group is serving)

Now, both IP pools have 65536 IP addresses.

Case of chunk size being too large Consider that addresses are divided into eight chunks to be distributed among five UPs. This means that not all UPs can get the comparable number of IP resources, and some UPs exhaust their IP resources sooner than the others.

Once IP resources reach exhaustion, UP override or load imbalance happens on such UPs. In this example, it can start happening after the usage of around 40960 addresses per pool or around 81920 addresses at IP pool group level.

- Pool1: (**UP1** = 8192 + 8192, **UP2** = 8192 + 8192, **UP3** = 8192 + 8192, **UP4**=8192, **UP5**=8192)
- Pool2: (**UP1** = 8192 + 8192, **UP2** = 8192 + 8192, **UP3** = 8192 + 8192, **UP4**=8192, **UP5**=8192)

Case of chunk size being too small: Consider same pools as described in the previous section are designed with chunk size of 512 with 128 chunks. In that case,

- Pool1: (**UP1** = 26 *512, **UP2** =26*512, **UP3** = 26*512, **UP4**=25*512, **UP5**=25*512)
- Pool2: (**UP1** = 26 *512, **UP2** =26*512, **UP3** = 26*512, **UP4**=25*512, **UP5**=25*512)

Although in this case, UP override or load imbalance due to chunk imbalance happens after 128000 (50 chunks*512 size*5 UPs) addresses is used, each UP has only 1k address per minute from each UP. Since the threshold timer is 60 sec, but incoming subscriber rate is 6k, each UP sees load imbalance for 5k subscribers.

Case of Correct design: Better design is to divide the IP pool into chunk size, for example, 4096 IP addresses. This design provides the CP with 16 chunks to distribute to five UPs.

Pool1: (**UP1** = 4096 + 4096 + 4096 + 4096, **UP2** = 4096 + 4096 + 4096, **UP3** = 4096 + 4096 + 4096, **UP4**= 4096 + 4096 + 4096, **UP5**= 4096 + 4096 + 4096)

In this case, UP override or load imbalance due to chunk imbalance happens after 122880 (six chunks *4096 size*5 UPs) addresses is used, and each UP has only 8k addresses per minute from each UP. Since threshold timer is 60 sec and incoming subscriber rate is 6k, UPs can cater to all subscribers.

As evident in this case, chunk size of 4096 results in better IP resource distribution among the UPs than having 8096 chunk size. As the IP resources reach exhaustion, the UP override or load imbalance happens on such UPs. In this example, it starts happening after the usage of around 61440 addresses per pool. Also, because there are two IP pools in the pool group at the same time, chunk replenishment is 4k from pool1 + 4k from pool2, which is greater than the required rate of 6k.

UP Grouping Guidelines

You must separate the consumer and enterprise customers in different UP groups since the IP pool size and routing requirements for each of them are different. Default UP group is used for the consumer customers. It's recommended to create specific User Groups for the enterprise customers and associate with enterprise APNs. Though the chunking mechanism provides the efficient IP address management for larger pools, it should be avoided in case of smaller pool sizes (for example, IP pool size less than 4k). It's recommended to dedicate the smaller IP pools to a particular UP. You can achieve this by having single UP in specific UP group and associating it to the APN.

Consider a case where you have five UPs and want to serve 40 small enterprise APNs of 256 addresses, and eight large consumer APNs of 64k addresses each. In this case, make two UP-groups, UPGroup1 consisting of one UP and 40 small enterprise APNs, and UPGroup2 consisting of four UPs and eight large consumer APNs.

UP Addition Guidelines

Before adding a new UP in a UP group, operator should ensure that there are free chunks available for all APNs at CP, which can be allocated during UP registration. This ensures that there are no uneven load distribution, as this UP still gets selected for call distribution even if one APN doesn't have any IP address available for this UP.

Miscellaneous Guidelines

- Use Pool Usage threshold alarms to get warning about the replenishment of IP pool resources. Take appropriate action, that is, adding a new pool in pool group.
- For IPv4v6 sessions in case of dynamic v4v6 address allocation, both IPv4 and IPv6 addresses that need to be assigned to UE belong to the same UP. IP pool planning must ensure that enough v4 and v6 IP addresses are available at per UP level on which v4v6 calls are expected.

Consider one IPv4 pool with name “poolv4” x.x.x.x/17 of 32k address and IPv6 pool with name “poolv6” x:x:x:x:/48 of 64k address. Consider that there are two APNs, APN1 which wants to make v4v6 call and operates on both “poolv4” and “poolv6”:

ip pool poolv4 209.165.200.224/17 – 32k address used by APN1

ipv6 pool poolv6 prefix 2003::/48 - 64k address used by APN2

If APN2 ends up using more than 32k IPv6 address, then there might not be sufficient IPv6 addresses left for v4v6 for APN1 due to poor planning of APNs. Segregate poolv6 of 64k address into poolv6_1 of 32k address used by APN1 and poolv6_2 of 32k address used by APN2. If needed, APN2 should add more IPv6 pools if it requires more IP addresses rather than starving APN1 of its fair share.

Static IP Pool Guidelines

Static IP pool is used in following scenarios:

- UE sends IP Address in Initial Attach.
- AAA/S6b returned IP Address.
- DHCP returned IP Address.

In the case of static IP pool, address is already decided by UE, and the benefit of UP selection goes away. Only the UP that has the chunk, which contains the selected IP address, can serve that call. For static IP pool, chunks are given to UP when UE requests first IP from an unused chunk. These chunks are given to UPs in a UP group in round robin fashion. Static chunks once allocated are never taken back from UP except in the case of Sx restart. The following are the guidelines with respect to static pools:

- For static IP pool, make one call as part of MOP procedure to dedicate that chunk to a particular UP. This helps in avoiding latency on first call, as the chunk is pushed on first call only.
- For static IP pool, use non-default UP group with limited number of UPs serving all the APNs with static IP pool.
- For static IP pool, use uniform chunk size across all pools. Pools should be of uniform size to avoid load imbalance on UPs, as UP selection can’t be used with static IP pool as explained before.
- For the static IPv4v6 PDNs to be successful, both IPv4 and IPv6 addresses need to be on the same UP. Only way to ensure this is to have a single UP in the UP group.
- For the multi-PDNs on same APN with one PDN static and other PDN dynamic to be successful, both addresses must be on the same UP. Since for dynamic pool, address is selected by UP selection algorithm, and for static address, UP is decided by the IP selected, the only way to avoid load imbalance is to have single UP in the UP group.