UCC 5G cnSGWc 構成および管理ガイド、リリース 2025.01

最終更新：2025 年 7 月 17 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（ **www.cisco.com/jp/go/safety_warning/** ）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

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.

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)

© 2025 Cisco Systems, Inc. All rights reserved.

目 次

はじめに：

このマニュアルについて xxxvii

使用する表記法 xxxvii

カスタマー サポートへの問い合わせ xxxviii

第 1 章

5G アーキテクチャ 1

概要 1

コントロール プレーン ネットワーク機能 1

ユーザー プレーン ネットワーク機能 2

サブスクライバ マイクロサービスのインフラストラクチャのアーキテクチャ 3

コントロール プレーン ネットワーク機能アーキテクチャ 4

第 2 章

cnSGW-C の概要 7

製品の説明 7

コンバージドコアの概要 7

使用例 8

展開アーキテクチャおよびインターフェイス 13

cnSGW-C アーキテクチャ 13

cnSGW-C 展開 13

コンバージド コア アーキテクチャ 15

コンバージドコア展開 16

サポートされるインターフェイス 17

データパケットのライフサイクル 17

ライセンス情報 17

標準準拠 17

第 3 章 オペレーションセンターでの cnSGW-C の展開と構成 19
機能の概要と変更履歴 19
要約データ 19
マニュアルの変更履歴 20
機能説明 20
cnSGW-C Ops センター 20
前提条件 21
cnSGW-C サービスの設定 21
ノードラベルを持つマッピングポッド 21
cnSGW-C の展開とアクセス 22
cnSGW-C の展開 22
cnSGW-C Ops センターへのアクセス 22
デイゼロ設定 23
Day 1 構成のロード 25
Day1config.cli 26

第 4 章 スマートライセンスサポート 43
機能の概要と変更履歴 43
要約データ 43
マニュアルの変更履歴 44
スマート ソフトウェア ライセンシング 44
Cisco Software Central 45
スマート アカウントとバーチャル アカウント 45
Cisco スマートアカウントの要求 45
cnSGW-C スマートライセンス 46
ソフトウェアタグと権限付与タグ 46
複数の権限付与タグ 47
機能説明 47
機能の仕組み 47
設定例 49

設定チェック 49
トラブルシューティング 50
スマート ライセンスの設定 51
Cisco Software Central にアクセスできるユーザー 51
Cisco Software Central にアクセスできないユーザー 55
スマートライセンス情報の表示 59

第 5 章 **cnSGW-C ローリング ソフトウェア アップデート 61**
機能の概要と変更履歴 61
要約データ 61
マニュアルの変更履歴 61
はじめに 62
cnSGW-C の更新 63
SMI Cluster Manager を使用したローリング ソフトウェア アップデート 64
前提条件 65
ローリング ソフトウェア アップグレードのトリガー 69
更新手順のモニタリング 71
ポッド詳細の表示 72
非 SMI クラスタでのローリング ソフトウェア アップデート 74
ローリングアップグレード最適化 75
機能説明 78
ローリングアップグレード最適化の仕組み 79
ポッドのアップグレード 79
ローリングアップグレード最適化サポートを利用したバージョンへのソフトウェアのアップグレード 81
制限事項 82
ローリングアップグレードでサポートされる機能の構成 82
ローリングアップグレード最適化の確認 83
OAM サポート 84
バルク統計 84

第 6 章 ポッドとサービスの参考資料 85
機能の概要と変更履歴 85
要約データ 85
マニュアルの変更履歴 85
機能説明 86
ポッド 87
UDP プロキシポッド 89
機能説明 89
サービス 91
オープンポートとサービス 93
ノードへのポッドの関連付け 95
ポッドの詳細とステータスの表示 96
ポッドの詳細 96
状態 96

第 7 章 3GPP RAN/NAS 原因コードのサポート 99
機能の概要と変更履歴 99
要約データ 99
マニュアルの変更履歴 99
機能説明 100
機能の仕組み 101
コール フロー 101
ベアラー作成手順のコールフロー 101
ベアラーの更新手順のコールフロー 102
ベアラー削除コマンドプロシージャのコールフロー 103
セッション削除プロシージャのコールフロー 105

第 8 章 アクセスベアラーのリリースサポート 107
機能の概要と変更履歴 107
要約データ 107

マニュアルの変更履歴 107
機能説明 108
機能の仕組み 108
コール フロー 108
アクセスベアラーのリリース（アクティブから IDLE トランザクション）コールフロー 108

第 9 章 APN プロファイルのサポート 111
機能の概要と変更履歴 111
要約データ 111
マニュアルの変更履歴 111
機能説明 112
機能設定 112
DNN プロファイルの構成 112
ネットワーク要素プロファイルの設定 112
設定の変更による影響 113
IP 送信元違反に対するアップリンクパケットの検証 115
IP 送信元違反に対するアップリンクパケットの検証の仕組み 116
IP 送信元違反に対するアップリンクパケットの検証の有効化と無効化 117
DNN プロファイルでのアップリンクパケット送信元検証の確認 118
NF サービスのアップリンクパケット送信元の検証の確認 119
トラブルシューティング情報 120
設定エラー 120

第 10 章 変更通知要求の処理 121
機能の概要と変更履歴 121
要約データ 121
マニュアルの変更履歴 121
機能説明 122
標準準拠 122
機能の仕組み 122

コール フロー 122
変更通知要求のコールフロー 122
OAM のサポート 124
バルク統計サポート 124

第 11 章 サブスクライバ要求のクリア 127
機能の概要と変更履歴 127
要約データ 127
マニュアルの変更履歴 127
機能説明 127
標準準拠 128
機能の仕組み 128
サポートされる clear コマンド 129
コール フロー 129
クリア PDN コールフロー 129

第 12 章 コンテキスト置換のサポート 133
機能の概要と変更履歴 133
要約データ 133
マニュアルの変更履歴 133
機能説明 134
機能の仕組み 134
コール フロー 134
完全なコンテキスト置換コールフロー 134
部分コンテキスト置換のコールフロー 135
OAM サポート 140
バルク統計 140

第 13 章 専用ベアラーのサポート 141
機能の概要と変更履歴 141
要約データ 141

マニュアルの変更履歴 141
機能説明 142
専用ベアラーの構成と更新 142
機能説明 142
機能の仕組み 142
コール フロー 142
専用ベアラーの削除 149
機能説明 149
機能の仕組み 150
コール フロー 150

第 14 章 削除ベアラーおよび削除セッション要求 153
機能の概要と変更履歴 153
要約データ 153
マニュアルの変更履歴 153
機能説明 153
MME から削除 154
PGW から削除 154
標準規格への準拠 154
機能の仕組み 155
コール フロー 155

第 15 章 ダウンリンクデータ通知 159
機能の概要と変更履歴 159
要約データ 159
マニュアルの変更履歴 160
機能説明 160
DDN メッセージ処理 160
機能説明 160
機能の仕組み 160
コール フロー 161

機能設定 168
DDN 障害タイマーの構成 168
DDN No User Connect Retry Timer の設定 168
設定例 169
設定の確認 169
制御メッセージでトリガされる DDN のサポート 169
機能説明 169
機能の仕組み 169
コール フロー 170
機能設定 171
設定例 171
設定の確認 171
DDN 制御手順の無効化 172
DDN Advance 機能 172
機能説明 172
機能の仕組み 172
コール フロー 172
標準準拠 179
機能設定 180
設定例 180
OAM のサポート 180
バルク統計 180

第 16 章
DSCP マーキングのサポート 183
機能の概要と変更履歴 183
要約データ 183
マニュアルの変更履歴 183
機能説明 184
データパケットの DSCP マーキング 184
機能説明 184
機能の仕組み 184

機能設定 185
設定例 186
設定の確認 186
CP シグナリングメッセージの DSCP マーキング 187
機能説明 187
機能設定 187
GTP エンドポイントの S11 インターフェイスでの DSCP の構成 187
GTP エンドポイントの S5e インターフェイスでの DSCP の構成 188
プロトコルエンドポイントの Sxa インターフェイスでの DSCP の構成 189
DSCP 構成の削除 189

第 17 章 BGP を使用したダイナミックルーティング 191
機能の概要と変更履歴 191
要約データ 191
マニュアルの変更履歴 191
機能説明 192
機能の仕組み 192
外部ネットワーク障害 194
Geo スイッチオーバー 195
内部ネットワーク障害 195
ローカルスイッチオーバー 196
リカバリおよびフェールバック 196
コール フロー 196
アクティブ/スタンバイモードでの着信トラフィックのルートの公開 196
シングルプロトコルポッド障害コールフロー 198
発信トラフィックのルートの学習コールフロー 199
BGP を使用したダイナミックルーティングの設定 201
モニタリングおよびトラブルシューティング 204

第 18 章 緊急コール サポート 209
機能の概要と変更履歴 209

要約データ 209
マニュアルの変更履歴 209
機能説明 210
制限事項 210
機能の仕組み 210
コール フロー 210
緊急セッション作成コールフロー 210
OAM サポート 212
バルク統計サポート 212

第 19 章 セッションの適切な終了のための拡張 PFCP 関連付け解除手順 215
PFCP 関連付けリリースを介した UPF から cnSGWc へのコール切断通知 216
UPF が開始した PFCP セッションのリリース 217
UPF によって開始された拡張 PFCP 関連付け解除 218
PFCP セッションリリース手順を開始するための EPFAR の有効化 219
バルク統計 219

第 20 章 拡張および非標準 QCI 値のサポートと検証 223
機能の概要と変更履歴 223
要約データ 223
マニュアルの変更履歴 223
機能説明 223
拡張および非標準 QCI 値の検証 224
VoLTE マーキングの拡張および非標準 QCI 値のサポートと検証 224

第 21 章 eMPS および WPS のサポート 225
機能の概要と変更履歴 225
要約データ 225
マニュアルの変更履歴 225
機能説明 226
eMPS/WPS サポート 226

機能説明 226
eMPS GTPv2 負荷/過負荷自己保護除外のサポート 226
機能説明 226
メッセージの優先順位プロファイル 228
機能説明 229
メッセージ優先順位プロファイルを選択する仕組み 230
機能設定 233
WPS プロファイルの設定 233
設定例 234
設定の確認 234
メッセージの優先順位プロファイルの設定 234
WPS プロファイルと SGW プロファイルの関連付けの構成 235
設定例 235
設定の確認 235
WPS プロファイルと DNN プロファイルの関連付けの設定 236
設定例 236
設定の確認 236
SGW QoS プロファイルの構成 236
sgw-qos-profile を sgw-profile および DNN プロファイルと関連付け 237
OAM のサポート 238
モニタリングおよびトラブルシューティング 238
バルク統計サポート 239

第 22 章 障害およびエラーの処理サポート 241
機能の概要と変更履歴 241
要約データ 241
マニュアルの変更履歴 242
概要 242
接続と接続解除の失敗とエラーの処理 242
Create Session Request の障害処理 243
デフォルトのベアラーの削除手順の失敗処理 244

セッション削除手順の障害処理 244
接続手順中のセッション セットアップ タイマー 245
ベアラー作成/更新/削除要求および応答の失敗とエラーの処理 246
ベアラー作成手順の失敗の処理 246
削除専用ベアラー手順の失敗処理 247
Update Bearer 手順の障害処理 248
無線アクセスベアラーや Modify Bearer Request の障害とエラーの処理 252
障害原因コード、原因ソース、ベアラーコンテキストエラーのサポート 254
障害原因コード 254
原因ソース 254
ベアラーコンテキストエラー 255

第 23 章 **GTPC および Sx パスの管理 257**
機能の概要と変更履歴 257
要約データ 257
マニュアルの変更履歴 258
機能説明 258
GTPC および Sx パスの管理 258
機能説明 258
機能設定 258
エコーパラメータの構成 259
ハートビートの設定 259
ピア構成の表示 260
設定例 261
OAM サポート 262
アラート 262
バルク統計サポート 262
GTPC パス障害 264
機能説明 264
機能の仕組み 264
GTPC パス障害検出 264

パスの障害の処理 265
機能設定 266
パス障害検出時のアクションの構成 266
ピアノードを更新する通知の設定 266
設定例 266
OAM のサポート 266
バルク統計サポート 267
Sx パス障害 267
機能説明 267
機能の仕組み 267
Sx パス障害の検出 268
パスの障害の処理 268
ハートビート処理 268
OAM のサポート 269
バルク統計サポート 269
パス障害検出のカスタマイズ 269
機能説明 269
機能設定 270
Sx パス障害のカスタマイズの設定 270
GTPC パス障害のカスタマイズの構成 270
OAM サポート 271
バルク統計サポート 271

第 24 章 GTPU エラーの表示 275
機能の概要と変更履歴 275
要約データ 275
マニュアルの変更履歴 275
機能説明 276
機能の仕組み 276
エラー表示のサポート 276
local-purge として s1u を使用するデフォルト ベアラー コール フロー 278

s1u を local-purge コールフローとして使用する専用ベアラー 279
s1u を local-purge コールフローとして使用する専用ベアラー（IDFT） 281
s1u を page-ue として使用するデフォルト/専用ベアラーコールフロー 283
local-purge/signal-peer として s5u を使用するデフォルト ベアラー コール フロー 285
local-purge/signal-peer として s5u を使用する専用ベアラーコールフロー 286
グレースフル終了 288
グレースフル終了のコールフロー 288
セッションの再配置 290
デフォルトベアラーのセッション置換コールフロー 291
専用ベアラーのセッション置換コールフロー 293
機能設定 295
設定例 296
構成の確認 296
OAM のサポート 296
バルク統計 296

第 25 章 GTPU パス障害 299
機能の概要と変更履歴 299
要約データ 299
マニュアルの変更履歴 299
機能説明 300
機能の仕組み 300
コール フロー 301
デフォルトのベアラーコールフローのパス障害 301
専用ベアラーのパス障害コールフロー 303
機能設定 305
設定例 306
設定の確認 306
GTPU パス障害 OAM のサポート 306
バルク統計 306

第 26 章 GTPv2 および Sx メッセージの再送信およびタイムアウトの処理 309
機能の概要と変更履歴 309
要約データ 309
マニュアルの変更履歴 309
機能説明 310
機能の仕組み 310
再送信とタイムアウトの値の設定 311
設定の確認 312

第 27 章 GTPv2 負荷/過負荷のサポート 315
機能の概要と変更履歴 315
要約データ 315
マニュアルの変更履歴 315
機能説明 316
GTPv2 負荷および過負荷機能の構成 317
負荷プロファイルの構成 318
設定例 319
過負荷除外プロファイルの設定 319
過負荷条件プロファイルの構成 320
最大セッション数の設定 321
設定例 322
過負荷プロファイルと SGW プロファイルの関連付け 322
設定例 324
設定の確認 325
負荷係数パラメータの構成 325
設定の確認 326
GTPv2 ロードおよびオーバーロード OAM のサポート 326
バルク統計 326

第 28 章 GTPv2 メッセージ検証 329

機能の概要と変更履歴 329
要約データ 329
マニュアルの変更履歴 329
機能説明 330
機能の仕組み 330
コール フロー 330
SGW 入力（S11）の基本検証と事前検証コールフロー 330
SGW 出力（S5）の基本検証と事前検証コールフロー 332

第 29 章 IDFT サポート 335
機能の概要と変更履歴 335
要約データ 335
マニュアルの変更履歴 335
機能説明 336
標準準拠 336
機能の仕組み 336
コール フロー 336
SGW 再配置を伴わない IDFT のサポートコールフロー 336
SGW 再配置を伴う IDFT のサポートコールフロー 338
Pure-S コールの 5G から 4G へのハンドオーバーのコールフロー 340
Pure-S コールの 4G から 5G へのハンドオーバーのコールフロー 342
IDFT（システムレベル）の作成コールフロー 343
IDFT（システムレベル）の削除コールフロー 345
OAM サポート 346
IDFT 構成の表示 346
障害処理 348
バルク統計サポート 350

第 30 章 アイドル セッション タイムアウトの構成 353
機能の概要と変更履歴 353
要約データ 353

マニュアルの変更履歴 353
機能説明 354
機能の仕組み 354
コール フロー 354
非アクティブ レポート コールフロー 355
UPF におけるアイドルタイマーの処理コールフロー 356
再アクティブ レポート コール フロー 358
ClearCall 処理コールフロー 359
機能設定 360
設定例 361
設定の確認 361

第 31 章 初期接続のサポート 363
機能の概要と変更履歴 363
要約データ 363
マニュアルの変更履歴 363
機能説明 364
機能の仕組み 364
コール フロー 364
初期接続コールフロー 364
標準準拠 367
バックオフタイマー、開始タイムスタンプ、および最大待機時間のサポート 367
バックオフ タイマー 368
発信タイムスタンプ 368
MaxWaitTime 368

第 32 章 システム間 RAT ハンドオーバー 369
機能の概要と変更履歴 369
要約データ 369
マニュアルの変更履歴 369
機能説明 370

機能の仕組み 370
コール フロー 370
Wi-Fi から LTE への成功コールフロー 370
OI インジケータの設定を使用した GnGp から LTE へのハンドオーバーコールフロー 372
OI インジケータの設定解除を使用した GnGp から LTE へのハンドオーバーコールフロー 374
標準準拠 376

第 33 章 MME 内および MME 間ハンドオーバー手順 377
機能の概要と変更履歴 377
要約データ 377
マニュアルの変更履歴 377
機能説明 378
機能の仕組み 378
コール フロー 378
MME 間ハンドオーバーのアクティブ-アクティブ移行コールフロー 378
MME 内ハンドオーバーのアクティブ-アクティブ移行コールフロー 379
MME 間/内ハンドオーバーのアイドル/アイドル移行コールフロー 381
MME 間/MME 内ハンドオーバーのアクティブ-アイドル移行コールフロー 382
MME 間ハンドオーバーおよびマルチ PDN 処理における、ULI 変更を伴うアクティブ-アイドル移行のコールフロー 383
削除コールフローとしてマークされたベアラーコンテキストを使用した MME 間ハンドオーバー 385
MME 内および MME 間ハンドオーバーの手順の OAM サポート 387
バルク統計 387

第 34 章 SGW サービスでの MCC/MNC 構成 389
機能の概要と変更履歴 389
要約データ 389
マニュアルの変更履歴 389
機能説明 390
機能の仕組み 390

コール フロー 390
PLMN タイプの検出コールフロー 390
SGW サービスでの MCC または MNC の設定 391
設定例 392
OAM サポート 392
バルク統計サポート 392

第 35 章 メッセージインタラクションのサポート 395
機能の概要と変更履歴 395
要約データ 395
マニュアルの変更履歴 396
機能説明 396
機能の仕組み 397
コール フロー 397
CBR Multi-PDN コールフロー 397
既存の PDN 手順のグレースフル停止のコールフロー 400
マルチ PDN 処理（PGW インタラクションあり）を使用した MME 間ハンドオーバー コールフロー 403
マルチ PDN コール X2 ハンドオーバー SGW の cnSGW-C への再配置コールフロー 405
サービスポッド（SGW）へのマルチ PDN S1 ハンドオーバー SGW の再配置コールフロー 408
同一 PDN に対する複数の CBR のコールフロー 412
コリジョンリゾルバの破棄処理コールフロー 414
一時停止処理コールフロー 416
優先度の低い手順の中止処理 419
二重削除最適化コールフロー 421

第 36 章 ベアラーコマンドサポートの変更と削除 423
機能の概要と変更履歴 423
要約データ 423
マニュアルの変更履歴 423

機能説明 424
機能の仕組み 424
コール フロー 424
MBC 障害処理コールフロー 424
MBC 成功処理コールフロー 426
DBC 障害処理コールフロー 427
DBC 成功処理コールフロー 429

第 37 章 ベアラー変更要求サポート 431
機能の概要と変更履歴 431
要約データ 431
マニュアルの変更履歴 431
機能説明 431
機能の仕組み 432
コール フロー 432
PGW インタラクションを伴わない UE トリガーサービス要求のコールフロー 432
PGW インタラクションを伴う UE トリガーサービス要求コールフロー 434

第 38 章 モニターサブスクライバおよびプロトコルサポート 437
機能の概要と変更履歴 437
要約データ 437
マニュアルの変更履歴 437
機能説明 438
機能設定 438
モニターサブスクライバの設定 438
設定例 439
モニタープロトコルの設定 461
設定例 462
トランザクションメッセージの設定 473
設定例 473
ログへのアクセス 478

第 39 章 複数 PDN の接続または接続解除手順 481
機能の概要と変更履歴 481
要約データ 481
マニュアルの変更履歴 481
機能説明 481
機能の仕組み 482
コール フロー 482
UE 要求 PDN 接続コールフロー 482
UE 要求または MME 要求の PDN 切断コールフロー 485
PGW 要求の切断コールフロー 487

第 40 章 パフォーマンス最適化のサポート 491
機能の概要と変更履歴 492
要約データ 492
マニュアルの変更履歴 495
機能説明 496
GTPC-EP から SGW サービスへの非同期 BG-IPC 497
機能説明 497
バッチ ID の割り当て、リリース、および調整のサポート 497
機能説明 497
機能の仕組み 497
機能設定 498
OAM サポート 499
バルク統計 499
キャッシュポッドの最適化 500
機能説明 500
CDL フラッシュ間隔およびセッション有効期限のチューニング設定 501
機能説明 501
機能設定 501
設定例 501

DDN コールフロー最適化 502
機能説明 502
機能の仕組み 502
コール フロー 502
機能設定 506
設定例 507
OAM のサポート 507
バルク統計 507
DDN タイムアウト設定 507
機能説明 507
機能設定 508
Ops センターを使用したドメインベースのユーザー認可 508
機能説明 508
機能の仕組み 508
機能設定 509
設定例 510
設定の確認 510
エッジエコーの実装 511
機能説明 511
機能の仕組み 511
OAM サポート 512
バルク統計サポート 512
ETCD ピア最適化サポート 513
機能説明 513
機能の仕組み 513
最適化された GTPv2 エンコーダおよびデコーダ 514
機能説明 514
機能設定 515
設定例 515
OAM サポート 516
バルク統計サポート 516

GR スプリットを使用した GTPC エンドポイント 516
機能説明 516
機能の仕組み 516
S11 と S5 による GTPC エンドポイント インターフェイス分割 517
機能説明 517
機能の仕組み 517
機能設定 519
設定例 519
GTPC IPC クロスラックサポート 520
機能説明 520
機能の仕組み 522
コール フロー 523
機能設定 528
設定例 529
OAM のサポート 529
KPI のサポート 529
サービス間ポッド通信 530
機能説明 530
機能の仕組み 530
コール フロー 531
OAM のサポート 533
統計のサポート 533
MBR コールフロー最適化 533
機能説明 533
機能の仕組み 534
コール フロー 534
機能設定 542
設定例 542
設定の確認 542
OAM サポート 543
バルク統計サポート 543

メンテナンス モード 544
機能説明 544
機能の仕組み 544
制限事項 545
メンテナンスモードの有効化または無効化 545
メンテナンスモードの有効化または無効化の例 545
メンテナンスモードの状態の確認 545
アイドル/アクティブの部分的な CDL 更新コールフロー 546
機能説明 546
機能の仕組み 546
制限事項 547
機能設定 547
設定例 547
OAM サポート 548
バルク統計サポート 548
DLDR スロットリングのサポートを含む PFCP セッションレポート 548
機能説明 548
機能の仕組み 549
機能設定 549
設定例 550
設定の確認 550
OAM サポート 550
バルク統計サポート 551
S11 インターフェイスでのセッション作成要求に対するスロットリングのサポート 552
機能説明 552
機能の仕組み 552
セッション作成要求に対するスロットリングの有効化 552
設定例 554
設定の確認 554
OAM サポート 554
バルク統計サポート 554

復元力処理 555
機能説明 555
機能の仕組み 556
機能設定 556
設定例 558
設定の確認 558
OAM のサポート 558
バルク統計サポート 558
ローミングピアパス管理の最適化 559
機能説明 559
機能の仕組み 559
機能設定 560
オペレータポリシーおよびサブスクライバポリシーの構成 560
設定例 561
デフォルト ゲートウェイの設定 562
設定例 562
設定の確認 563
ローミングピアノードのエコー要求最適化の有効化 563
OAM のサポート 567
バルク統計サポート 567
DB データベース更新にフラグを設定する 568
機能説明 568
OAM サポート 568
バルク統計サポート 568
プロトコルマイクロサービスに統合された UDP プロキシ機能 569
機能説明 569
UDP プロキシバイパスを使用した PFCP プロトコルエンドポイント 569
UDP プロキシバイパスを使用した PFCP プロトコルエンドポイント 570

第 41 章 **Presence Reporting Area** 575
機能の概要と変更履歴 575

要約データ 575
マニュアルの変更履歴 575
機能説明 575
機能の仕組み 576

第 42 章 冗長性サポート 579
機能の概要と変更履歴 579
要約データ 579
マニュアルの変更履歴 579
高可用性のサポート 580
機能説明 580
UDP プロキシの高可用性 580
アーキテクチャ 580
cnSGW-C ポッドおよび VM の展開レイアウト 580
機能の仕組み 581
ポッドレベルのラベル付けとレプリカの構成 582
設定例 582
設定の確認 583
ラック間冗長性サポート 583
機能説明 584
機能の仕組み 584
概要 584
ラック間冗長性トリガー 585
ラック NF ロール 586
一般的な注意事項 587
インスタンス認識 588
ラック間冗長性インスタンスの構成 588
エンドポイントインスタンス認識の構成 589
プロファイル cnSGW-C インスタンス認識の構成 591
cnSGW-C エンドポイントの設定 591
ダイナミック ルーティング 593

BGP を使用したダイナミックルーティングの設定 595
BGP スピーカーの設定 599
IPAM 599
IPAM の構成 601
Geo レプリケーション 601
ETCD/CachePod レプリケーションの設定 603
Geo モニタリング 604
ポッドモニタリング 604
リモート クラスタ モニタリング 605
トラフィック モニタリング 605
BFD モニタリング 606
CDL GR の展開 609
CDL GR の前提条件 609
CDL インスタンス認識とレプリケーション 609
合法的傍受 614
RADIUS 設定 615
GR ペアのソフトウェアアップデート 616
GR CLI 620
Geo ロールの切り替え 621
Geo ロールのリセット 621
トラブルシューティング 622
show/clear コマンド 622
モニターサブスクライバ 633
モニタープロトコル 633
地理的冗長性 OAM のサポート 634
RMA 手順の前提条件 634
ヘルス チェック 635
回復手順 639
主要業績評価指標（KPI） 641
バルク統計 648
アラート 650

メンテナンス モード 657

第 43 章 サービス構成の拡張機能 659

機能の概要と変更履歴 659

要約データ 659

マニュアルの変更履歴 659

機能説明 660

機能設定 660

SGW プロファイルの設定 660

設定例 660

設定の確認 661

サブスクライバポリシーの構成 661

設定例 662

オペレータポリシーの設定 662

設定例 663

ポリシー DNN の設定 663

設定例 664

設定の変更による影響 664

トラブルシューティング情報 666

設定エラー 666

第 44 章 SGW 変更サポート 667

機能の概要と変更履歴 667

要約データ 667

マニュアルの変更履歴 667

機能説明 668

アーキテクチャ 669

ローミングのサポート 670

機能の仕組み 670

コール フロー 670

初期接続での URR のインストールコールフロー 670

SGW CDR コールフロー 672
切断時の URR の削除と CDR レポートのコールフロー 674
しきい値に達した使用状況レポートのコールフロー 676
専用ベアラーの URR インストールのコールフロー 678
専用ベアラーの削除時の URR の削除と CDR の生成コールフロー 679
S11 トリガーに関するボリュームレポートのコールフロー 681
S5 トリガーに関するボリューム レポート コールフロー 683
標準準拠 685
制限事項 685
機能設定 686
CLI 設定 686
cnSGW-C 課金プロファイルまたは GTP プライムの構成 687
課金モードの設定 693
cnSGW-C 課金しきい値の設定 693
cnSGW-C 課金しきい値と cnSGW-C 課金プロファイルの関連付けの構成 694
コール制御プロファイルの構成 696
呼制御プロファイルでの課金特性の設定 697
表示 CLI 698
GTPP-EP SFTP Push CLI 698
cnSGW-CDR でサポートされている CDR フィールド 698
custom24 ディクショナリ 699
custom24 におけるフィールドの ASN.1 定義 708
SGW 変更 OAM サポート 716
バルク統計 716

第 45 章
SGW 再配置のサポート 721
機能の概要と変更履歴 721
要約データ 721
マニュアルの変更履歴 721
機能説明 722
機能の仕組み 722

コール フロー 722
cnSGW-C への X2 ハンドオーバー SGW 再配置コールフロー 722
cnSGW-C への S1 ハンドオーバー SGW 再配置コールフロー 724
cnSGW-C への TAU X2 ハンドオーバー SGW 再配置コールフロー 727
CN-SGW（マルチ PDN）への X2 ハンドオーバー SGW 再配置のコールフロー 728
CN-SGW（マルチ PDN）への S1 ハンドオーバー SGW 再配置コールフロー 731
ベアラーコンテキストが削除対象としてマークされた状態での X2 ハンドオーバー SGW 再配置のコールフロー 734
ベアラーコンテキストが削除対象としてマークされた状態での S1 ハンドオーバー SGW 再配置のコールフロー 736
ベアラーコンテキストの数が少ない場合の MME 間および MME 内のハンドオーバーと S1 SGW 再配置コールフロー 740
SGW リロケーション OAM サポート 741

第 46 章
Sx 負荷/過負荷制御の処理 743
機能の概要と変更履歴 743
要約データ 743
マニュアルの変更履歴 743
機能説明 744
機能の仕組み 744
ノード機能のサポート 744
UP の選択 745
ピア過負荷シナリオ中の Sx メッセージの処理 745
Sx Establishment に対するスロットリングのサポート 745
自己保護モードでのユーザープレーンからのセッション終了トリガー 745
輻輳原因に対する障害処理プロファイル サポート 746
Sx 負荷/過負荷機能の設定 746
障害処理プロファイルの構成 747
自己過負荷状態での Sx 要求メッセージのスロットリングと除外 749
Sx 負荷/過負荷制御 OAM のサポート 750
バルク統計 750

第 47 章 古いセッションの処理とクリア 753
機能の概要と変更履歴 753
要約データ 753
マニュアルの変更履歴 753
機能説明 754
機能の仕組み 754
コール フロー 754
タイマー期限切れ処理のコールフロー 754
機能設定 757
設定例 757
設定の確認 757
OAM サポート 757
バルク統計 757

第 48 章 CSFB 手順のサポート：中断と再開 759
機能の概要と変更履歴 759
要約データ 759
マニュアルの変更履歴 759
機能説明 760
機能の仕組み 760
コール フロー 761
Suspend Notification コールフロー 761
再開通知のコールフロー 762
再開用の空のベアラー変更要求のコールフロー 764

第 49 章 ベアラー更新要求および応答 767
機能の概要と変更履歴 767
要約データ 767
マニュアルの変更履歴 767
機能説明 768

標準準拠 768
機能の仕組み 768
コール フロー 768

第 50 章 UPF 選択のサポート 773
機能の概要と変更履歴 773
要約データ 773
マニュアルの変更履歴 774
機能説明 774
DNN および DCNR サポートを使用した UPF の選択 774
機能説明 774
機能の仕組み 774
UPF の選択方法 775
UPF 選択方式の構成 776
UPF グループプロファイルベースの UPF 選択の設定 776
ネットワークベースの UPF 選択の設定 777
ポリシーベースの UPF 選択の設定 778
トラブルシューティング情報 779
設定エラー 779
ロケーションサポートを使用した UPF の選択 779
機能説明 779
UPF 選択機能の設定 779
EPS 用の ECGI の構成 779
TAI グループの設定 781
Location-area-group の設定 782
UPF グループと UPF 選択ポリシーの機能拡張の構成 783
cnSGW-C と SMF の UPF の組み合わせ選択 784
機能説明 784
標準準拠 784
機能の仕組み 784
システム アーキテクチャ 785

コール フロー 786
cnSGW-C と SMF の複合 UPF 選択の構成 793
コンバージド コア プロファイルの構成 794
ノード ID の設定 795
UPF 選択 OAM のサポート 795
バルク統計 795

第 51 章 VoLTE コールの優先順位付け 797
機能の概要と変更履歴 797
要約データ 797
マニュアルの変更履歴 797
機能説明 798
機能の仕組み 798
機能設定 798
優先順位の構成 799
設定例 799
設定の確認 799
Sx メッセージの優先順位 801
OAM サポート 801
バルク統計 802

第 52 章 cnSGW-C のトラブルシューティング 803
説明 803
CLI データの使用 803
show subscriber および cdl show コマンド 804
ログ 808
イベント障害のログ 808
機能の仕組み 811
イベントロギングの有効化 813

第 53 章 cnSGW-C 構成例 815

設定例 815

このマニュアルについて

（注） この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。既存の偏った用語が置換される間は、製品ソフトウェアのユーザーインターフェイスにハードコードされている言語、RFP のドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

ここでは、『ウルトラ クラウド サービング ゲートウェイ コントロール プレーン機能 - 構成および管理ガイド』、ドキュメントの表記法、およびカスタマーサポートの詳細について説明します。

- 使用する表記法 （xxxvii ページ）
- カスタマー サポートへの問い合わせ （xxxviii ページ）

使用する表記法

次の表に、このマニュアル全体で使用される表記法を示します。

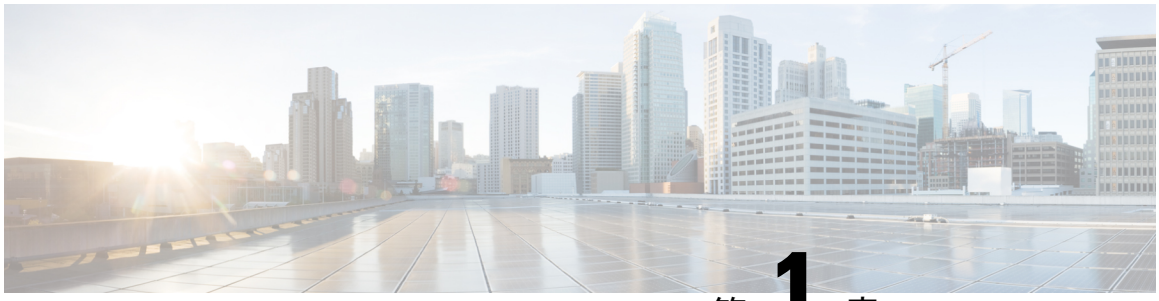
通知タイプ	説明
情報メモ	重要な機能または手順に関する情報を提供します。
注意	プログラム、デバイス、またはシステムに損傷を与えるおそれがあることを注意喚起します。
警告	人身傷害または死亡事故のおそれがあることを警告します。また、電気的障害のおそれがあることを警告する場合もあります。

書体の表記法	説明
スクリーンディスプレイとして表されるテキスト	この書体は、端末画面に表示されるディスプレイを表します。次に例を示します。 ログイン：
commands として表現されるテキスト	この書体は、入力したコマンドを表します。次に例を示します。 **show ip access-list** このマニュアルでは、コマンドの完全表記に常に小文字を使用しています。コマンドには、大文字と小文字の区別はありません。
command *variable*として表されるテキスト	この書体は、コマンドの一部である変数を表します。次に例を示します。 **show card** *slot_number* *slot_number* は、該当するシャーシのスロット番号を表す変数です。
メニュー名またはサブメニュー名として表されるテキスト	この書体は、ソフトウェアアプリケーション内でアクセスするメニューとサブメニューを表します。次に例を示します。 [File] メニュー、[New] の順にクリックてください。

カスタマー サポートへの問い合わせ

カスタマーサポートに連絡するには、この項の情報を使用してください。

最新の製品マニュアルについては、http://www.cisco.com のサポートエリアを参照するか、サービスリクエストを送信してください。このサイトにアクセスするには、有効なユーザー名とパスワードが必要です。詳細については、シスコの販売担当者またはサービス担当者にお問い合わせください。



第 1 章

5G アーキテクチャ

- 概要 （1 ページ）
- サブスクライバ マイクロサービスのインフラストラクチャのアーキテクチャ （3 ページ）
- コントロール プレーン ネットワーク機能アーキテクチャ （4 ページ）

概要

Ultra Cloud Core は、5G New Radio（NR）Standalone （SA）モードの 3GPP 標準規格をサポートするシスコのソリューションです。これらの標準規格では、ネットワークのパフォーマンスと機能を向上させるために、コントロールプレーン（CP）とユーザープレーン（UP）機能（CUPS など）の分離に基づいて、さまざまなネットワーク機能（NF）が定義されます。

コントロール プレーン ネットワーク機能

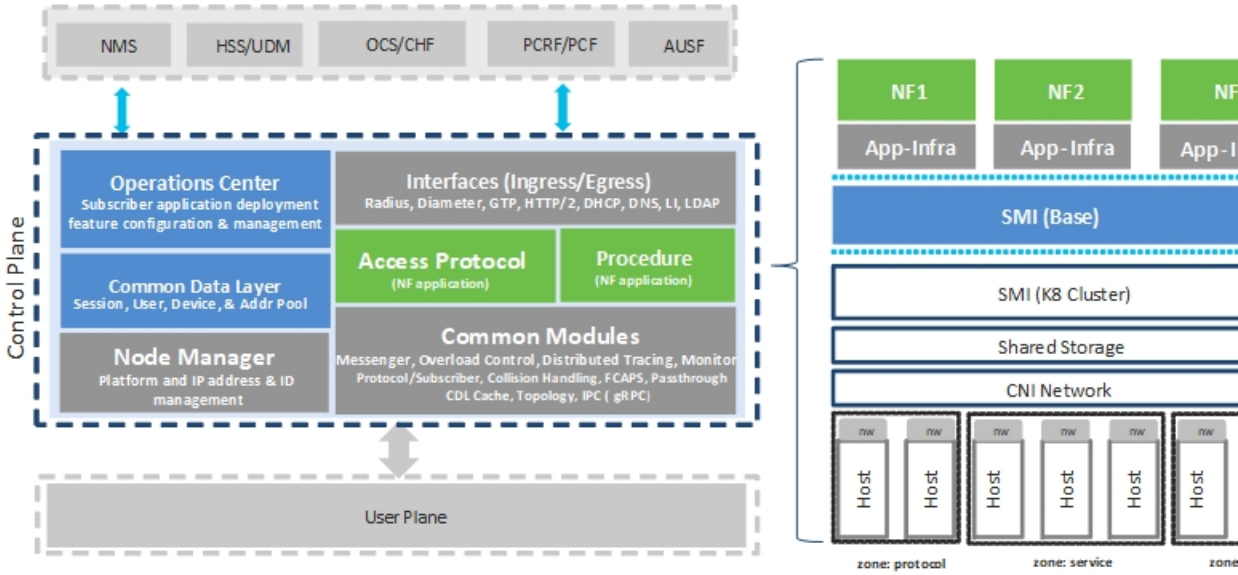
Ultra Cloud Core を構成する CP 関連の NF は、次のテナントを中心に設計された共通のアーキテクチャに基づいています。

- クラウドスケール：完全に仮想化され、シンプルさ、スピード、柔軟性を実現します。
- 自動化とオーケストレーション：最適化された運用、サービスの構築、およびインフラストラクチャ。
- セキュリティ：インフラストラクチャから NF アプリケーションまで、展開スタック全体における複数のセキュリティ層。
- API 公開：オープンかつ広範で、可視性、制御、およびサービスイネーブルメントが向上します。
- アクセスに依存しない：複数のネットワークタイプをサポート（5G、4G、3G、Wi-Fi など）。

これらのコントロールプレーン NF はそれぞれ、サブスクライバ マイクロサービス インフラストラクチャ（SMI）を介した展開用のコンテナ化されたアプリケーション（マイクロサービスなど）として設計されています。

SMIは、ライフサイクル管理（LCM）、運用と管理（OAM）、パッケージングなど、NFの機能面に対する共通アプリケーション層を定義します。

図 *1 : Ultra Cloud Core CP* アーキテクチャコンポーネント



ユーザー プレーン ネットワーク機能

Ultra Cloud Core 内の 5G UP NF は、ユーザープレーン機能（UPF）です。CP 関連の NF とは異なり、5G UPF は、Cisco 4G CUPS アーキテクチャ内のユーザー プレーン コンポーネントで現在使用されているのと同じ Vector Packet Processing（VPP）テクノロジーを利用しています。この共通性により、4G と 5G の間で次のような一貫した機能のセットが提供されます。

- 超高速パケット転送。
- 広範に統合された IP サービス（サブスクライバ ファイアウォール、テザリング、ディープ パケット インスペクション（DPI）、Internet Content Adaption Protocol（ICAP）、アプリケーションの検出および制御（ADC）、ヘッダーエンリッチメント（HE）など）。
- トラフィックおよびTCPを最適化するための統合されたサードパーティ製アプリケーション。

サブスクライバ マイクロサービスのインフラストラクチャのアーキテクチャ

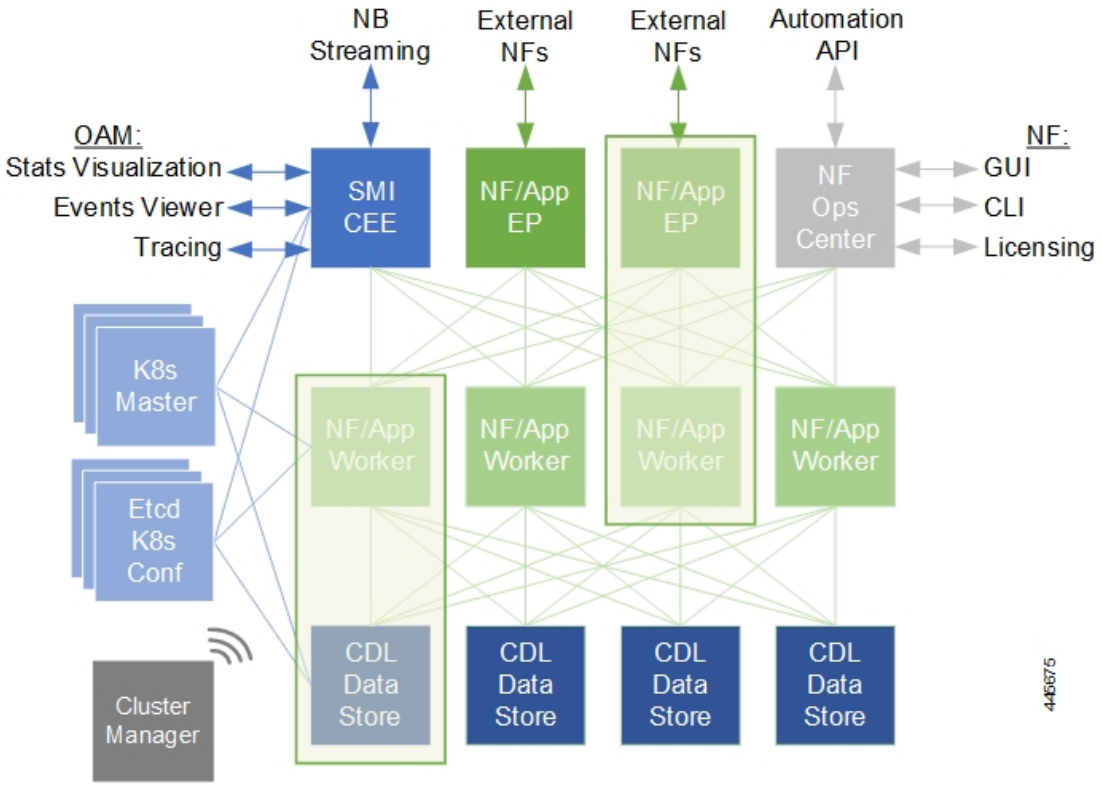
Ultra Cloud Core Subscriber Microservices Infrastructure（SMI）は、マイクロサービスベースのアプリケーションの迅速な展開とシームレスなライフサイクル運用を可能にするクラウドテクノロジーの階層化されたスタックです。

SMI スタックは次のもので構成されています。

- SMI クラスタマネージャ：Kubernetes（K8s）クラスタを作成し、ソフトウェアリポジトリを作成して、展開、アップグレード、拡張を含むクラスタの継続的なLCMを提供します。
- Kubernetes 管理：クラスタに展開される NF アプリケーションに LCM を提供する K8s プライマリおよびetcd機能が含まれます。このコンポーネントは、クラスタヘルスモニタリングとリソーススケジューリングも提供します。
- 共通実行環境（CEE）：ライセンス付与と権限付与の機能、設定管理、テレメトリとアラームの可視化、ロギング管理、障害対応ユーティリティなど、Cisco クラウドネイティブ NF とアプリケーションの共通ユーティリティおよび OAM 機能を提供します。また、これらのツールと展開されたアプリケーションに関連するすべてのお客様との接触ポイントと統合ポイントで、一貫した対話とエクスペリエンスを提供します。
- Common Data Layer（CDL）：5Gおよびサブスクライバアプリケーション向けに特別に設計された、高性能で低遅延のステートフルなデータストアを提供します。この次世代データストアは、ローカルまたは地理的冗長性の展開で高可用性を提供します。
- サービスメッシュ：アプリケーションコンテナ間の高度なメッセージルーティングを提供し、管理された相互接続性、追加のセキュリティ、および低リスクの方法で新しいコードと新しい構成を展開する機能を実現します。
- NB ストリーミング：課金および課金システムに「ノースバウンド データ ストリーミング」サービスを提供します。
- NF またはアプリケーション ワーカー ノード：NF アプリケーションポッドを構成するコンテナ。
- NF またはアプリケーション エンドポイント（EP）：NF またはアプリケーションと、それらのネットワーク上の他のエンティティへのインターフェイス
- アプリケーションプログラミングインターフェイス（API）：展開、構成、および管理の自動化のためのさまざまな API を提供します。

次の図は、これらのコンポーネントがどのように相互接続し、マイクロサービスベースの NF またはアプリケーションを構成するかを示しています。

図 *2 : SMI* コンポーネント



SMI コンポーネントの詳細については、『Ultra Cloud Core サブスクライバ マイクロサービス インフラストラクチャ』および『展開ガイド』の「概要」の章にある関連ドキュメントを参照してください。

コントロールプレーンネットワーク機能アーキテクチャ

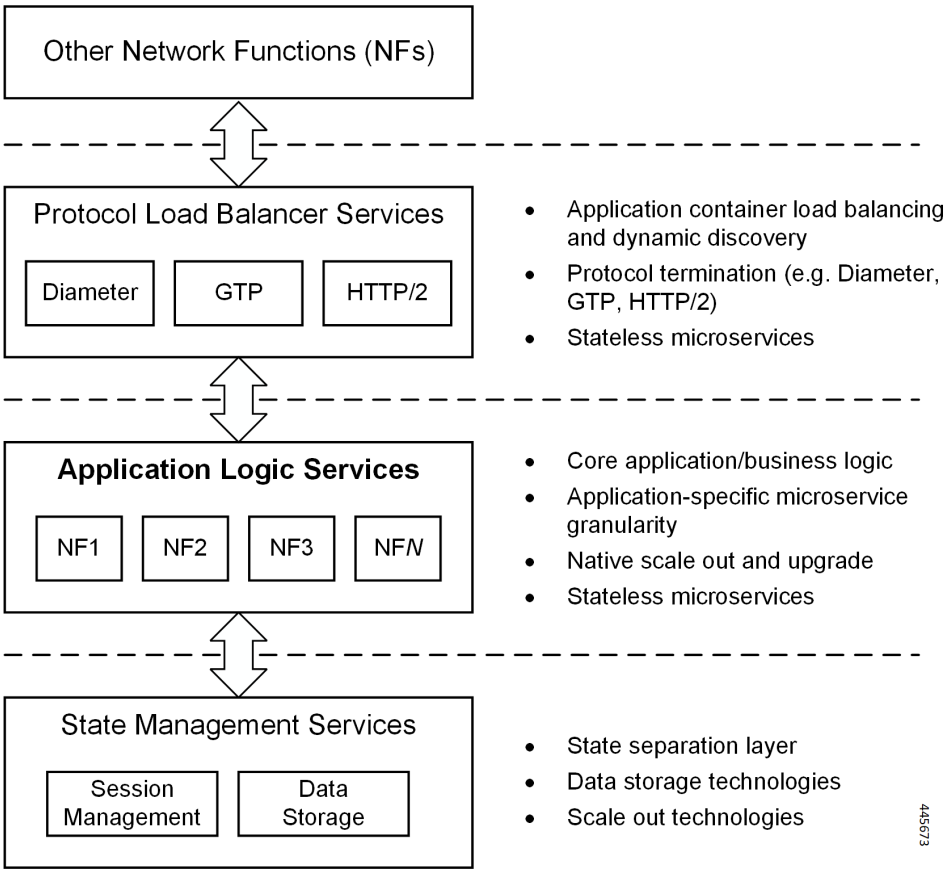
コントロールプレーン（CP）NF は、クラウドネイティブ環境内で提供されるステートフルまたはステートレス機能を利用する 3 階層アーキテクチャに基づいて設計されています。

アーキテクチャ層は次のとおりです。

- プロトコル ロード バランサ サービス：このサービスは、アプリケーションコンテナの動的な検出と、プロトコルのプロキシと終了を主な役割とするステートレスマイクロサービスです。対象には、従来の 3GPP プロトコルと、5G で導入された新しいプロトコルが含まれます。
- アプリケーションサービス：コアアプリケーションまたはビジネスロジックの実装を行い、受信した情報に基づいて実際のアプリケーションをレンダリングするステートレスサービスです。この層には、さまざまなレベルのマイクロサービスの粒度が含まれる場合があります。アプリケーションサービスはステートレスです。

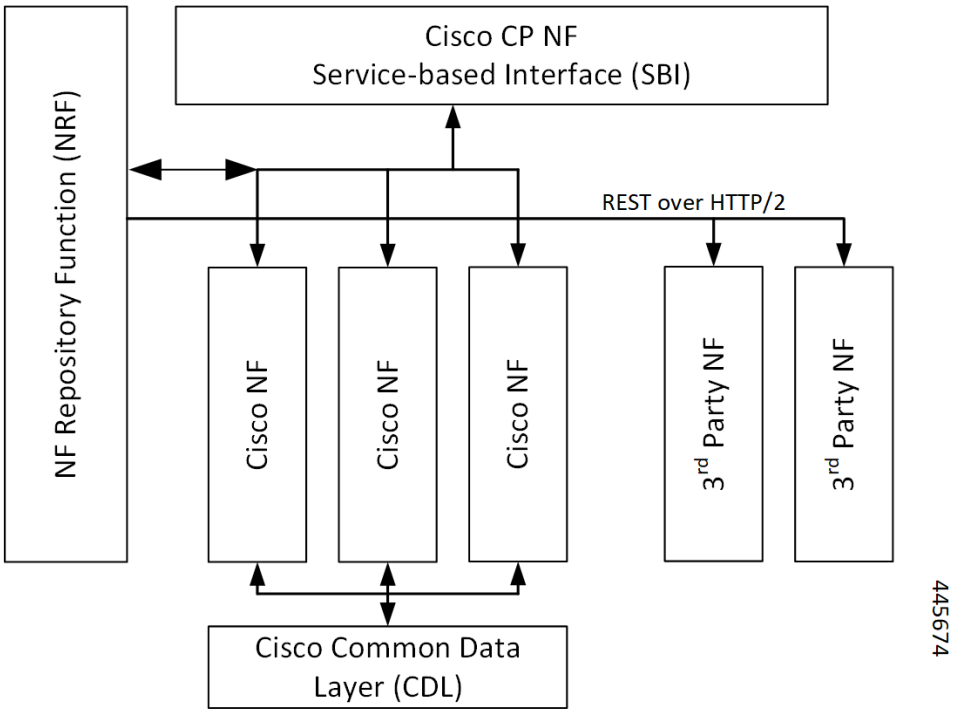
- ステート管理サービス：ステートレス アプリケーション サービスは、セッションデータやサブスクライバデータなどのステート情報を保存またはキャッシュするための共通データ層（CDL）を提供します。この層は、インメモリキャッシュから本格的なデータベースまで、さまざまなデータ ストレージ テクノロジーをサポートします。

図 *3 :* コントロール プレーン ネットワーク機能階層型アーキテクチャ

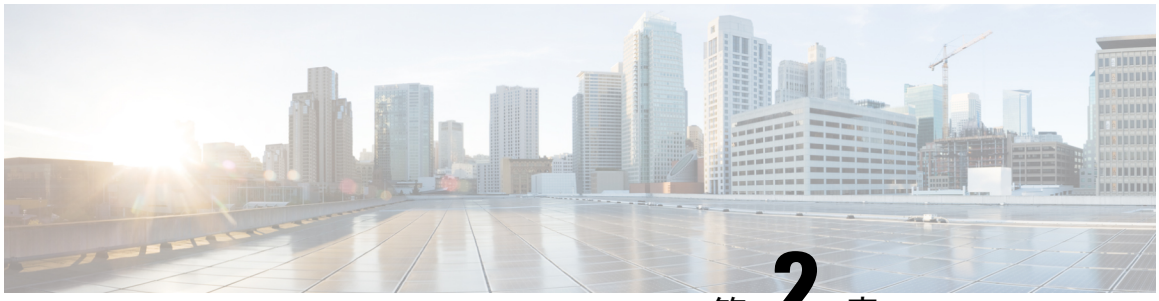


Cisco CP NF が設計されている 3 階層アーキテクチャは、3GPP で定義されている 5G コア（5GC）のサービスベース アーキテクチャ（SBA）を完全にサポートしています。Cisco CP NF は、3GPP の定義に従い、HTTP/2 over TCP を使用して、サービスベースのインターフェイス（SBI）を介して相互に通信したり、サードパーティ NF と通信したりします。

図 *4 : Cisco CP NF* サービスベース アーキテクチャのサポート



シスコのネットワーク機能の詳細については、対応するネットワーク機能のマニュアルを参照してください。



第 2 章

cnSGW-C の概要

- 製品の説明 （7 ページ）
- コンバージドコアの概要 （7 ページ）
- 使用例 （8 ページ）
- 展開アーキテクチャおよびインターフェイス （13 ページ）
- データパケットのライフサイクル （17 ページ）
- ライセンス情報 （17 ページ）
- 標準準拠 （17 ページ）

製品の説明

cnSGW-C は、コンバージド コア ネットワーク（4G-5GC）のコントロール プレーン ネットワーク機能（NF）です。サービング ゲートウェイ コントロールプレーン機能（cnSGW-C）は、SMI アーキテクチャ上に構築されます。cnSGW-C は UE アンカーとして機能し、3GPP TS 23.401 および 3GPP TS 23.214 で規定されているセッションのセットアップおよび終了手順とともに、モビリティ手順をサポートします。

サービング ゲートウェイ コントロールプレーン機能（cnSGW-C）は、3GPP 仕様 23.214 の表 4.3.2-1 に記載されているように、SGW-U によって実行される機能を除き、TS 23.401 [2] で定義されている S-GW の機能を提供します。さらに、cnSGW-C は SGW-U を選択し（3GPP 仕様 23.214 の 4.3.3 節を参照）、TS 23.214 の表 4.3.2-1 に記載されている機能に関して SGW-U を制御します。

Cisco Cloud Native プラットフォームに基づく SMF（IWF）サポートにより、ハードウェア使用率と O&M アクティビティを向上させるために、Cloud Native プラットフォームで cnSGW-C 機能をサポートすることが推奨されます。

コンバージドコアの概要

コンバージド コア ソリューションは、4G および 5G デバイス、およびユースケースをサポートする機能を備えた、クラウドネイティブの高度なコンバージド コントロール プレーンを提供します。

重要 このリリースでは、S5C および cnSGW-C 機能を備えたクラウドネイティブ統合 S-GW および SMF インスタンスのみがサポートされています。

コンバージド コア ソリューションは、あらゆるタイプのサブスクライバとユースケースを処理するコンバージド コア ネットワークを提供することで、運用の複雑さを軽減します。

オペレータには、次の利点があります。

- 4G サブスクライバまたは 5G から 4G カバレッジエリアへのハンドオフを処理しながら、cnSGW-C と SMF 間のシグナリングを減らすことで、全体的なネットワーク効率を向上させます。
- 4G カバレッジエリア内のサブスクライバの追加ホップ SGW-U が原因で発生する遅延を削減し、統合 UPF のデータパスを折りたたむことで、全体的なユーザー体験を向上します。
- 4G および 5G デバイスの SBA インターフェイスを使用して、統一されたサブスクライバ ポリシーと課金インフラストラクチャを使用する機能を提供します。

このソリューションは、次のコンバージド コントロール プレーンおよびユーザープレーン機能をサポートしています。

- コンバージド コントロール プレーン機能
  - S-GW および SMF ネットワーク機能を単一の Kubernetes 名前空間の下で単一の展開として統合し、E-UTRAN/NR（コンバージド コア ゲートウェイ）から 4G および 5G デバイスをサポートします。
  - 論理ネットワーク機能（データ）をサポート
- 統合ユーザープレーン機能
  - UPF 機能と SGW-U 機能を単一のネットワーク機能として統合
  - N4 インターフェイスと Sxa インターフェイスの同時サポートを提供
  - 単一の展開で複数のコントロールプレーンを終端

使用例

ここでは、cnSGW-C がサポートするユースケースについて説明します。

- **cnSGW-C 構成**

  cnSGW-C の基本構成は、cnSGW-C を動作させるために必要な構成の詳細なビューを提供します。構成には、cnSGW-C を展開するためのインフラストラクチャの設定、SMI を介した cnSGW-C の展開、および長期的に cnSGW-C の機能を利用するための Ops センター

の構成が含まれます。SMI の詳細については、『Ultra Cloud Core SMI クラスタデプロイヤ操作ガイド』を参照してください。

次の機能は、このユースケースに関連しています。

- APN プロファイルのサポート （111 ページ）
- サービス構成の拡張機能 （659 ページ）
- UPF 選択のサポート （773 ページ）

コンバージドコア展開の場合、cnSGW-C はコンバージド Ops センターを使用して展開されます。

- **セッション管理**

EPC にアクセスするすべての UE は、単一の S-GW に関連付けられます。cnSGW-C は、特定の UE に対して複数の PDN をサポートします。セッション管理の一環として、cnSGW-C は次をサポートします。

- デフォルトおよび専用ベアラーの確立
- ベアラーの変更
- ベアラーの非アクティブ化

次の機能は、このユースケースに関連しています。

- 3GPP RAN/NAS 原因コードのサポート （99 ページ）
- 変更通知要求の処理 （121 ページ）
- コンテキスト置換のサポート （133 ページ）
- 専用ベアラーのサポート （141 ページ）
- 削除ベアラーおよび削除セッション要求 （153 ページ）
- CP シグナリングメッセージの DSCP マーキング （187 ページ）
- eMPS および WPS のサポート （225 ページ）
- 緊急コール サポート （209 ページ）
- アイドル セッション タイムアウトの構成 （353 ページ）
- 初期接続のサポート （363 ページ）
- 複数 PDN の接続または接続解除手順 （481 ページ）
- Presence Reporting Area （575 ページ）
- ベアラー更新要求および応答 （767 ページ）
- VoLTE コールの優先順位付け （797 ページ）

- **UE モビリティのサポート**

  cnSGW-C は、UE のモビリティアンカーポイントです。LTE ネットワークでは、MME の変更の有無にかかわらず、eNodeB から eNodeB へのモビリティを発生させることができます。UE は、異なるモード、S1 ベースの再配置、X2 ベースの再配置、5G-4G インターワーキングにより 1 つの cnSGW-C から別の cnSGW-C に移動することもできます。

  次の機能は、このユースケースに関連しています。

  - IDFT サポート （335 ページ）
  - MME 内および MME 間ハンドオーバー手順 （377 ページ）
  - ベアラー変更要求サポート （431 ページ）
  - Presence Reporting Area （575 ページ）
  - SGW 再配置のサポート （721 ページ）

- **S1 - リリース/バッファリング/ダウンリンクデータ通知**

  cnSGW-C は、eNodeB と SGW-U 間の S1-U ベアラーのリリースを処理します。 cnSGW-C は、S1-U ベアラーが解放されたことを示す無線アクセスベアラー（RAB）メッセージを受信すると、ユーザープレーンを更新し、UE を IDLE 状態に移行します。IDLE 状態のときに、UE がダウンリンク データ パケットを受信すると、cnSGW-C は MME 向けの DDN メッセージをページ UE に生成します。

  cnSGW-C は、DDN スロットリング、DDN 遅延、および DDN の高優先度機能もサポートしています。

  次の機能は、このユースケースに関連しています。

  - アクセスベアラーのリリースサポート （107 ページ）
  - ダウンリンクデータ通知 （159 ページ）
  - DDN Advance 機能 （172 ページ）

- **再送信およびタイムアウト**

  すべての手順について、3GPP TS 23.401/29.274 に従って、cnSGW-C は N3 再送信および T3 タイムアウトサポートをサポートします。これらは、S11、S5、および Sx インターフェイスでサポートされています。

  次の機能は、このユースケースに関連しています。

  - GTPv2 および Sx メッセージの再送信およびタイムアウトの処理 （309 ページ）

- **失敗とエラーの処理**

  cnSGW-C は、次の処理をサポートしています。

  - 初期接続手順および追加の PDN 設定手順の一部としてのセッション作成要求に対する失敗応答

- PGW によって開始される専用ベアラー作成（DBC）手順の失敗のシナリオ
- 無線アクセスベアラー（RAB）、PGW およびユーザープレーンからのベアラー変更要求と応答（MBR）

次の機能は、このユースケースに関連しています。

- 障害およびエラーの処理サポート （241 ページ）

- **負荷/過負荷制御機能**

cnSGW-C は以下をサポートします。

- Sx インターフェイスを介した、ピアノードが過負荷状態である間の負荷/過負荷制御情報とアクションの交換。
- GTPv2 インターフェイスでの負荷/過負荷情報の処理。

次の機能は、このユースケースに関連しています。

- GTPv2 負荷/過負荷のサポート （315 ページ）
- Sx 負荷/過負荷制御の処理 （743 ページ）

- **cnSGW-C 課金サポート**

cnSGW-C は以下をサポートします。

- オフライン課金（Gz）。
- ローカルディスクストレージへの CDR の書き込み。CDR ファイルは定期的に SFTP サーバーにプッシュされます。
- 選択したサブスクライバの CDR の生成。これは、コール制御プロファイルを介してオペレータポリシーごとに CDR の生成を有効にすることで実現されます。

次の機能は、このユースケースに関連しています。

- SGW 変更サポート （667 ページ）

- **GTPC および Sx のピアとパスの管理**

cnSGW-C は以下をサポートします。

- MME（S11 ピア）、PGW（S5 ピア）、およびユーザープレーンのピア管理。
- ECHO 要求/応答およびハートビート要求/応答によるピアモニタリング。
- S11 および S5 ピアのパス障害イベントの処理。

次の機能は、このユースケースに関連しています。

- GTPC および Sx パスの管理 （258 ページ）
- GTPC パス障害 （264 ページ）

- パス障害検出のカスタマイズ （269 ページ）
- Sx パス障害 （267 ページ）

- **冗長性サポート**

  K8 クラスタでの cnSGW-C の導入は、高可用性（HA）と地理的冗長性（GR）をサポートする上で重要な役割を果たします。

  冗長性サポートは、ラックまたはクラスタの障害時にクラスタ間のステートフルセッションの継続性を保証します。

  cnSGW-C は、シングルポイント障害を回避するための各クラスタコンポーネントの冗長セットアップにより HA を実現します。

  GR は、ラック全体に、2 つの個別の K8 クラスタ間でデータを複製するためのラックレベルの冗長性を提供します。ラック/クラスタで障害が発生すると、トラフィックはリモートラックに切り替わり、トラフィックが処理されます。この障害は、電源障害、マルチコンピューティング障害、ネットワーク障害、マルチポッド障害、BFD リンク障害などが原因である可能性があります。

  次の機能は、このユースケースに関連しています。

  - 冗長性サポート （579 ページ）
  - 高可用性のサポート （580 ページ）

- **ダイナミック ルーティング**

  ダイナミックルーティングは、L2 スタティックルーティングに加えて、リーフとの L3 ピアリングを可能にします。

  次の機能は、このユースケースに関連しています。

  - BGP を使用したダイナミックルーティング （191 ページ）

- **GTPU パス管理とセッション管理**

  UPF 機能は、存在しない TEID を持つ GTP-PDU を受信したときに、GTP-U ピアのエラー表示メッセージを送信者に通知します。この通知により、古いセッションやベアラーがないことを確認でき、また、ネットワーク内の一貫性が維持されます。

  S-GW ノードと UPF ノード間のエラー表示および GTP-U パス障害表示の通信は、N4 インターフェイスを介してサポートされています。ネイバーノードの場合、通信は S1u/S5u インターフェイスを介してサポートされます。この実装では、エラー表示と GTPU パス障害の local-purge または signal-peer の動作のバリエーションが考慮されます。

  次の機能は、このユースケースに関連しています。

  - GTPU エラーの表示 （275 ページ）
  - GTPU パス障害 （299 ページ）

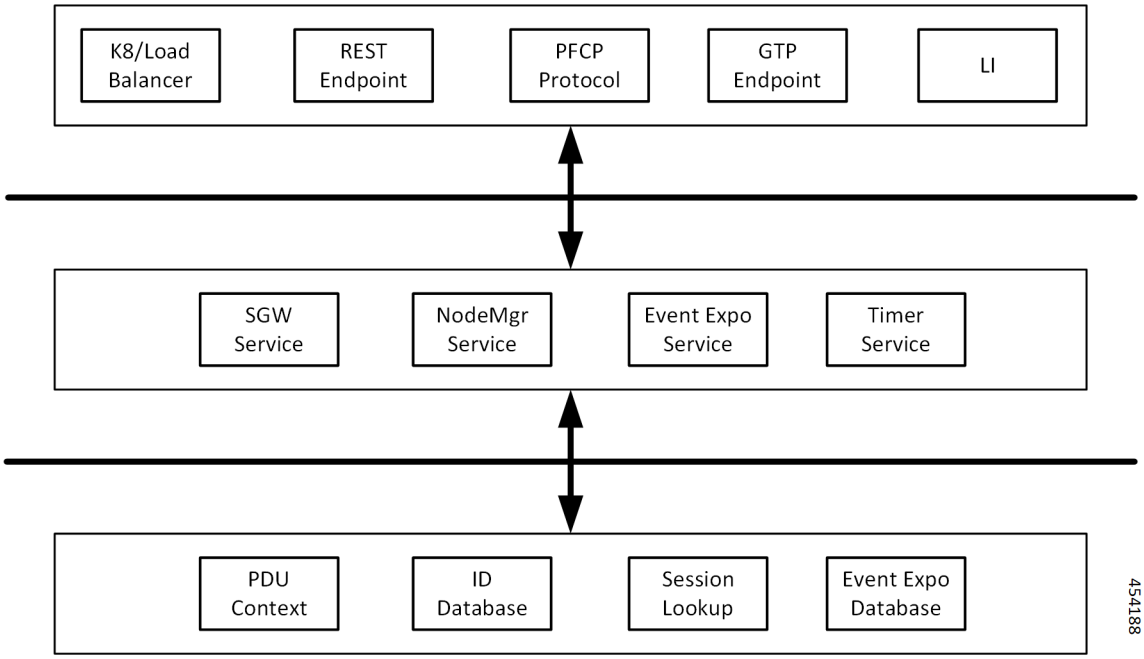
展開アーキテクチャおよびインターフェイス

cnSGW-C は、共通のモバイル コア プラットフォーム アーキテクチャを使用するコンバージド コア ネットワーク機能ポートフォリオの一部です。コアネットワーク機能には、アクセスおよびモビリティ管理機能（AMF）、ポリシー制御機能（PCF）、セッション管理機能（SMF）、およびユーザープレーン機能（UPF）が含まれます。

cnSGW-C アーキテクチャ

cnSGW-C ネットワーク機能は、疎結合のマイクロサービスで構成されます。マイクロサービスの分解は、次の図に示す 3 層のアーキテクチャに基づいています。

図 *5 : cnSGW-C* アーキテクチャ



次に、cnSGW-C アーキテクチャの 3 つのレイヤを示します。

- レイヤ 1 : プロトコルおよびロードバランササービス（ステートレス）
- レイヤ 2 : アプリケーションサービス（ステートレス）
- レイヤ 3 : データベースサービス（ステートフル）

cnSGW-C 展開

cnSGW-C NF は、独立した NF として別の名前空間に展開されます。

図 *6 : cnSGW-C* 展開

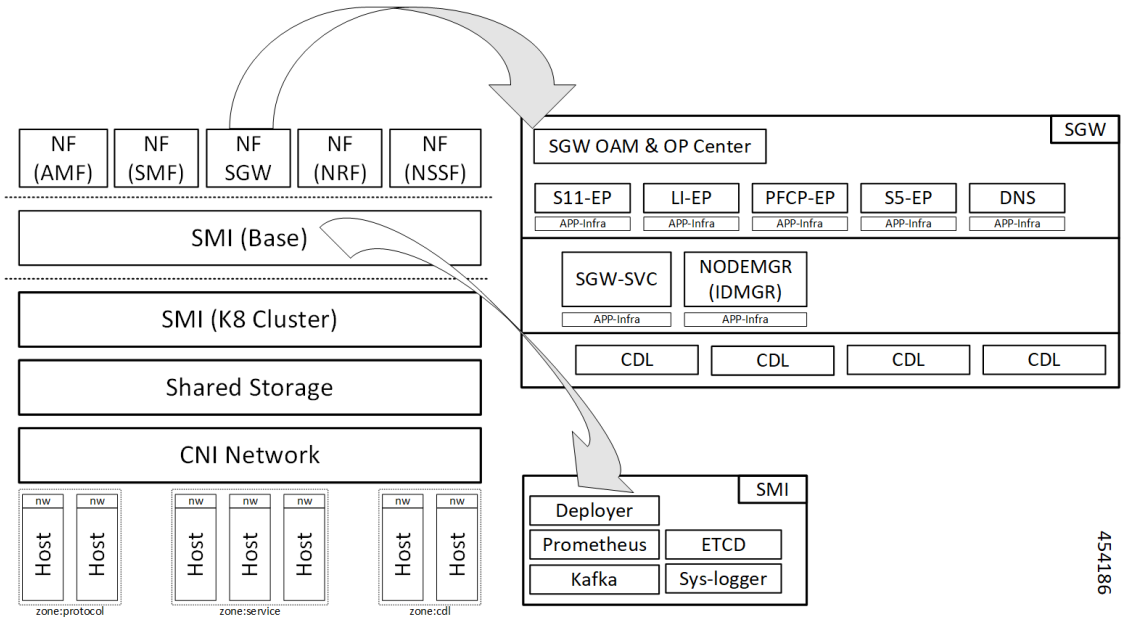
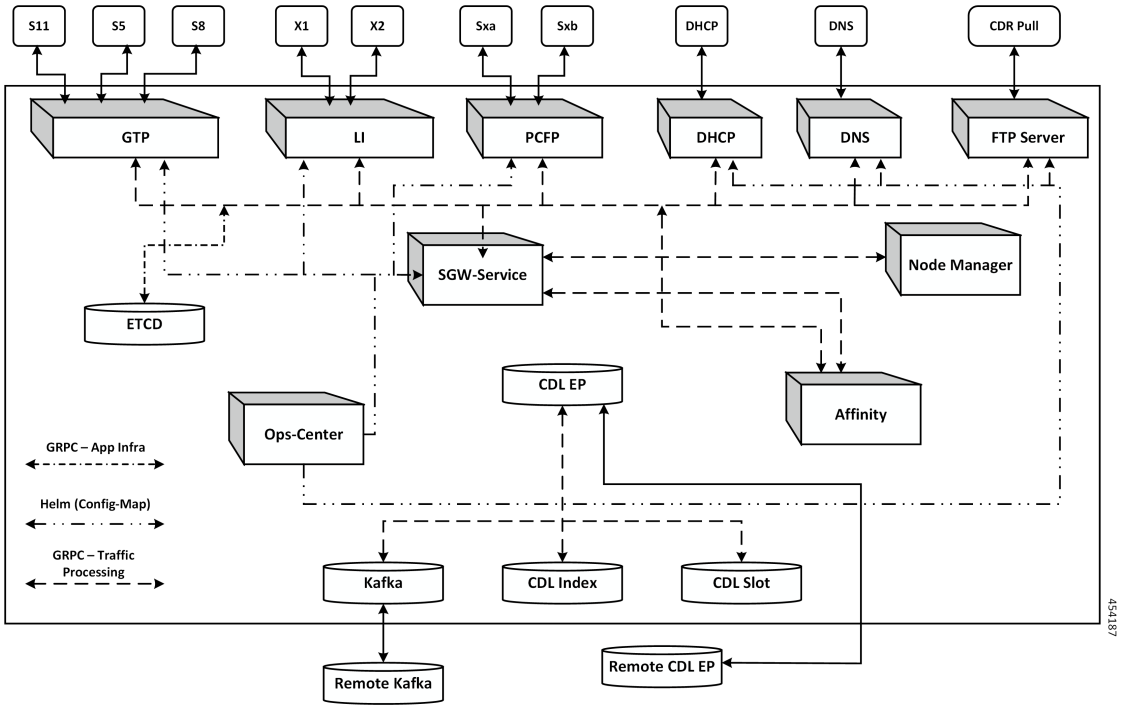


図 *7 : cnSGW-C HELM* チャート



コンバージド コア アーキテクチャ

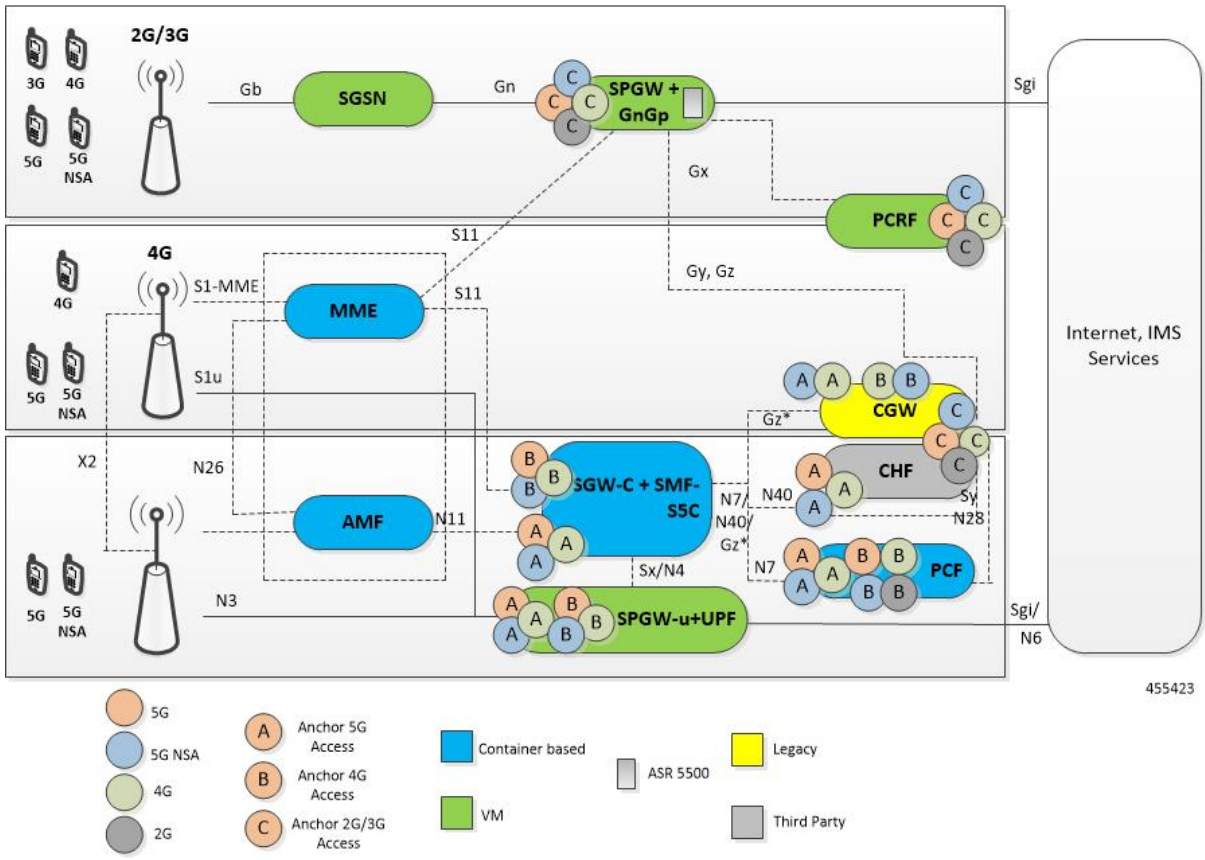
コンバージドコア ソリューションは、SMI アーキテクチャに基づく単一の統合プラットフォームを提供します。サポートアーキテクチャにより、クラウドネイティブの S-GW および SMF の展開が 5GC および cnSGW-C 機能と統合されます。このソリューションでは、ポリシーおよび課金機能に 3GPP 定義の SBA インターフェイスが使用されます。

コンバージド コア アーキテクチャでは、4G および 5G 対応 UE は、同じコントロール プレーン インスタンスにアンカーされます。コントロール プレーン インスタンスは、SMF、5GC、および cnSGW-C 機能を提供します。

4G と 5G アクセスタイプ間のハンドオフは、5G 対応デバイスではシームレスです。LTE から UTRAN（4G/5G と 3G/2G 間の双方向通信）へのハンドオフは、4G 対応デバイスについてはシームレスではありません。

次の図に、サポートされるネットワークアーキテクチャを示します。

図 *8 :* コンバージド コア アーキテクチャ



このソリューションの一部として展開される UPF は VPC-SI VM です。UPF の展開は VM ベースであり、以下をサポートしています。

- 同じインスタンス内の SGW-U、PGW-U、および UPF 機能、およびコントロールプレーンへの Sxa、Sxb、Sxab、または N4 インターフェイスの公開。

• 同時に複数の CP インスタンス（最大 4）。

コンバージドコア展開

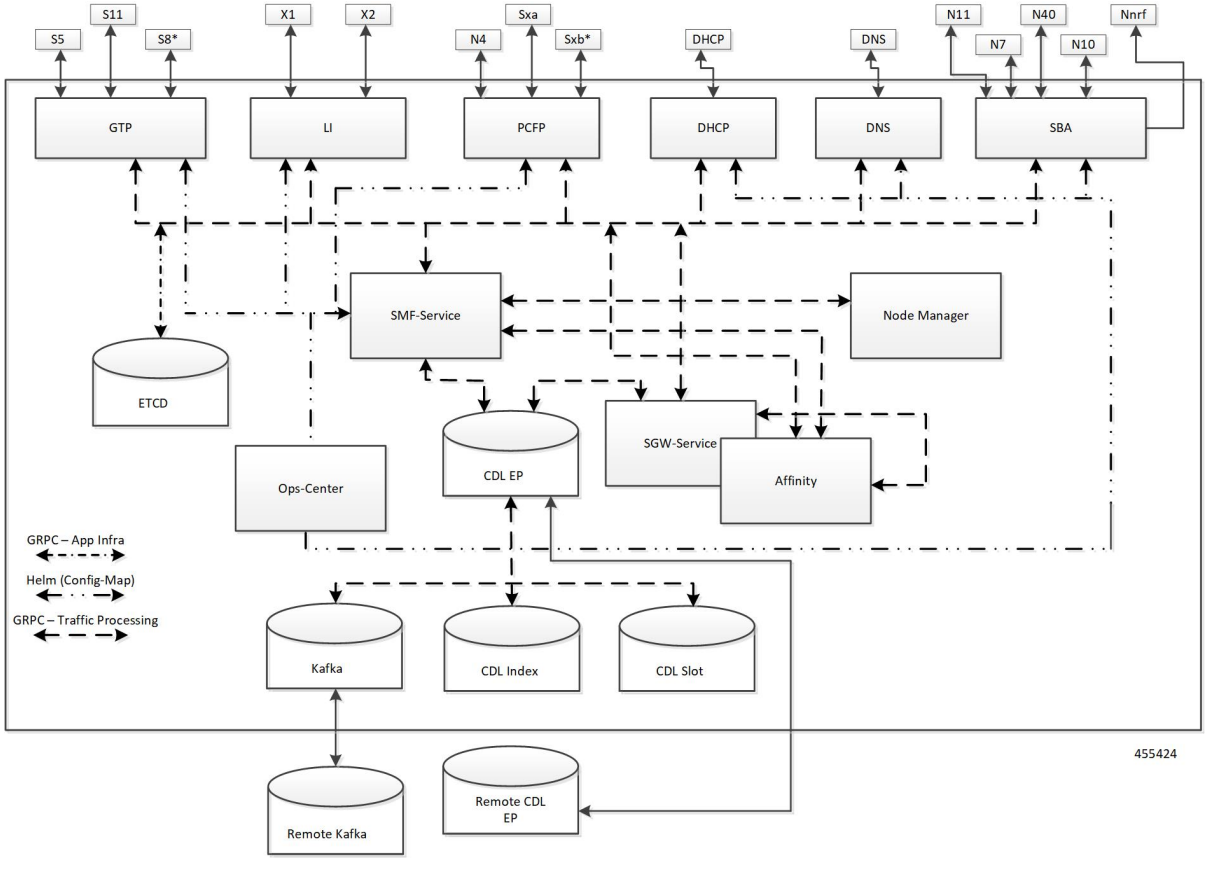
コンバージドコアの展開は、すべてのユースケースに対応するコンバージド コントロール プレーンと統一ユーザー プレーン インフラストラクチャに基づいています。

コンバージドコアの展開では、すべての 4G および 5G 対応 UE は 5G コア（SMF）にアンカーされ、PCF に対しては SBA インターフェイスが使用されます。

コンバージドコア展開には、cnSGW-C や SMF サービスをはじめとする各種マイクロサービスの設定を可能にするコンバージド Ops センターが含まれています。コンポーネントのインストールには、単一の製品 Helm チャートが使用されます。

次の図は、コンバージド S-GW および SMF ネットワーク機能の Kubernetes 展開を示しています。

図 *9 : Kubernetes* 展開



プロトコル層サービスは SMF と S-GW 全体で共有されます。GTP エンドポイントは、S11 インターフェイスと S5/S8 インターフェイスを終了します。同様に、PFCP（プロトコル）エンドポイントは、N4 インターフェイスと Sxa インターフェイスを終了します。

SMF および S-GW サービスは個別のポッドとして展開され、セッションの処理は分離されます。両方のサービスポッドは、サブスクライバセッションの保存に CDL を使用します。

サポートされるインターフェイス

ここでは、cnSGW-C と 5GC の他のネットワーク機能との間でサポートされるインターフェイスについて説明します。

- S11：SGW と MME 間の基準点
- S5/S8：SGW と PGW/SMF 間の基準点
- Sxa：SGW-C と SGW-U 間の基準点
- Gz：SGW-C と課金サーバーの間の基準点

データパケットのライフサイクル

データパケットのライフサイクルについては、「初期接続のサポート （363 ページ）」を参照してください。

ライセンス情報

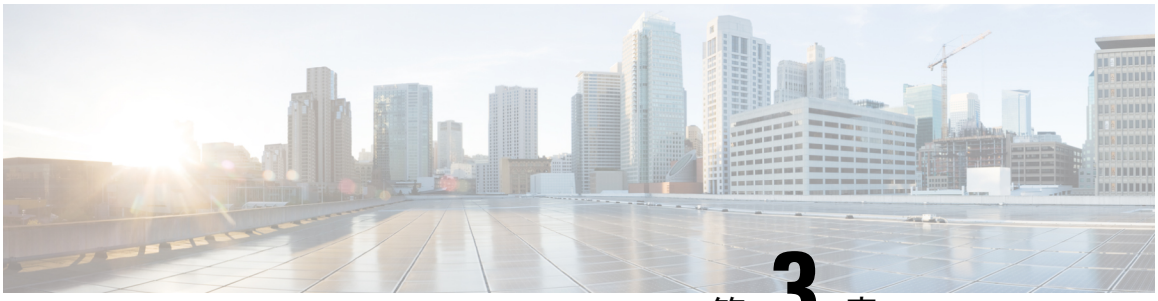
cnSGW-C は、Cisco Smart Licensing をサポートしています。詳細については、スマートライセンスサポート （43 ページ）を参照してください。

標準準拠

cnSGW-C は、次の 3GPP 標準規格に準拠しています。

- 3GPP TS 23.401「General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access」
- 3GPP TS 23.402「Architecture enhancements for non-3GPP accesses」
- 3GPP TS 29.274「Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C)」
- 3GPP TS 23.214「Architecture enhancements for control and user plane separation of EPC nodes」
- 3GPP TS 29.244「Interface between the Control Plane and the User Plane nodes」
- 3GPP TS 24.008「Mobile radio interface Layer 3 specification; Core network protocols; Stage 3」
- *3GPP TS 23.007*「*Restoration procedures*」

- *3GPP TS 22.153「Multimedia priority service」*
- *3GPP TS 33.107「3G security; Lawful interception architecture and functions」*



第 3 章

オペレーションセンターでの cnSGW-C の展開と構成

- 機能の概要と変更履歴 （19 ページ）
- 機能説明 （20 ページ）
- cnSGW-C サービスの設定 （21 ページ）
- cnSGW-C の展開とアクセス （22 ページ）
- Day 1 構成のロード （25 ページ）

機能の概要と変更履歴

要約データ

表 *1*: 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *2:* マニュアルの変更履歴

改訂の詳細	リリース
次の拡張機能が導入されました。 • 複数の利用資格タグ • ベアメタルサーバーでの cnSGW-C の展開	2021.02.0
最初の導入。	2020.07.0

機能説明

cnSGW-C の展開プロセスには、サブスクライバ マイクロサービス インフラストラクチャ（SMI）クラスタデプロイヤを介した cnSGW-C の展開が含まれます。Confd CLI に基づく cnSGW-C Ops センターを使用して、構成やカスタマイズを実行できます。

cnSGW-C Ops センター

Ops センターは、次の操作を実行するためのユーザーインターフェイスを提供する、システムレベルのインフラストラクチャです。

- 可変の Helm チャートパラメータを指定して、マイクロサービスの展開をトリガーします。これらのチャートパラメータは、展開に関連付けられている Kubernetes オブジェクト（展開、ポッド、サービスなど）のスケールとプロパティを制御します。
- Kubernetes の構成マップを介して、アプリケーション固有の構成を 1 つ以上のマイクロサービスにプッシュします。
- アプリケーション固有の実行コマンド（show コマンドや clear など）を発行します。これらのコマンドにより以下が実行されます。
  - アプリケーション固有のポッドで API を呼び出す
  - アプリケーションから返された情報をユーザーインターフェイスで表示する

次のスクリーンショットは、Web ベースの CLI の例です。

図 *10 : Web* ベースの *Ops* センター

cnSGW-C Ops センターを使用すると、ライセンス、cnSGW-C エンジン、EGPT および PFCP エンドポイント、CDL などの機能を構成できます。

前提条件

SMI レイヤに cnSGW-C を展開する前に、次の手順を実行します。

- すべての仮想ネットワーク機能（VNF）が展開されていることを確認します。
- cnSGW-C Ops Center と Cloud Native Common Execution Environment（CN-CEE）の SMI 同期操作を実行します。

cnSGW-C サービスの設定

cnSGW-C サービスには、Call Setup、Modify、および Delete Request を処理するための基本設定が必要です。

ノードラベルを持つマッピングポッド

前提条件

- ノードラベルがポッドの展開レイアウトに従っていることを確認します。
- 外部 VIP が NF の要件に従っていることを確認します。
- ポッド間トラフィックのロードバランスのために Istio を有効にします。

ノードラベルは、クラスタ同期でノードにアタッチされるキーと値のペアです。各ノードには、キーと値のセットのラベルを定義することができます。各キーはノードに対して一意である必要があります。ラベルを使用すると、ユーザーは NF ポッドを疎結合でノードにマッピングできます。

重要

- ポッドレベルのラベル付け設定は、cnSGW-C がベアメタルサーバーに展開されている場合にのみ適用されます。
- ポッドをマッピングする前に、必ず SMI クラスタデプロイヤでノードラベルを設定してください。以下は、マスター 1 ラベル付けのコマンド例です。

```
[cndp-clpnc-cm-cm-primary] SMI Cluster Deployer (config-nodes-master-1)# k8s
node-labels smi.cisco.com/svc-type smf-node
```

ポッドをノードラベルにマッピングするには、次の設定例を使用してください。

```
config
   k8 label protocol-layer key label_key value label_value
   k8 label service-layer key label_key value label_value
   k8 label cdl-layer key label_key value label_value
   k8 label oam-layer key label_key value label_value
   end
```

以下は、ポッドからノードラベルへのマッピングの例です。

```
k8 label protocol-layer key smi.cisco.com/node-type value smf-proto
exit
k8 label service-layer key vm-type value smf-svc
exit
k8 label cdl-layer key smi.cisco.com/node-type value smf-cdl
exit
k8 label oam-layer key smi.cisco.com/node-type value oam
exit
```

cnSGW-C の展開とアクセス

ここでは、cnSGW-C を展開する方法と、cnSGW-C Ops センターにアクセスする方法について説明します。

cnSGW-C の展開

サブスクライバマイクロサービスインフラストラクチャ（SMI）プラットフォームは、cnSGW-C アプリケーションとその他のネットワーク機能の展開および管理を担当します。

ベアメタルサーバー（現在は Cisco UCS-C サーバー）環境に cnSGW-C Ops センターを展開する方法については、『*Ultra Cloud Core* サブスクライバ マイクロサービス インフラストラクチャ：オペレーションガイド』の「ベアメタル上の *SMI Cluster Manager* の運用」の項を参照してください。

cnSGW-C Ops センターへのアクセス

SSH または Web ベースの CLI コンソールを介して cnSGW-C Ops センターに接続できます。

- SSH：

ssh admin@*ops_center_pod_ip* **-p 2024**

- Web ベースのコンソール
  1. Kubernetes マスターノードにログインします。
  2. 次のコマンドを実行します。

     kubectl get ingress <*namespace*>

     使用可能な入力接続が一覧表示されます。
  3. 適切な入力を選択し、Ops センターにアクセスします。
  4. Web ブラウザから次の URL にアクセスします。

     cli.<*namespace*>**-ops-center**.<*ip_address*>.**nip.io**

デフォルトでは、Day 0 の構成が cnSGW-C にロードされます。

デイゼロ設定

Day 0 構成を表示するには、次のコマンドを実行します。

show running-config

次に、Day 0 構成の例を示します。

```
system mode shutdown
helm default-repository base-repos
helm repository base-repos
 url https://charts.209.165.201.1.nip.io/ccg.2021.01.0.i60
exit
k8s name          2nd-a18-kub-cluster
k8s namespace     cn-cn3
k8s nf-name       smf
k8s registry      docker.209.165.201.1.nip.io/ccg.2021.01.0.i60
k8s single-node   false
k8s use-volume-claims false
k8s ingress-host-name 209.165.201.2.nip.io
k8s nodes 2nd-a18-kub-cluster-master-11
 node-type   master
 worker-type master
exit
k8s nodes 2nd-a18-kub-cluster-master-22
 node-type   master
 worker-type master
exit
k8s nodes 2nd-a18-kub-cluster-master-33
 node-type   master
 worker-type master
exit
aaa authentication users user admin
 uid        1117
 gid        1117
 password   $1$XNGJOr.C$iZZvQbNfmPN15qG4GpQa8/
 ssh_keydir /tmp/admin/.ssh
 homedir    /tmp/admin
exit
aaa ios level 0
 prompt "\h> "
```

```
exit
aaa ios level 15
 prompt "\h# "
exit
aaa ios privilege exec
 level 0
  command action
  exit
  command autowizard
  exit
  command enable
  exit
  command exit
  exit
  command help
  exit
  command startup
  exit
 exit
 level 15
  command configure
  exit
 exit
exit
nacm write-default deny
nacm groups group LI
 user-name [ liadmin ]
exit
nacm groups group admin
 user-name [ admin ]
exit
nacm rule-list admin
 group [ admin ]
 rule li-deny-tap
  module-name       lawful-intercept
  path              /lawful-intercept
  access-operations *
  action            deny
 exit
 rule li-deny-clear
  module-name       tailf-mobile-smf
  path              /clear/lawful-intercept
  access-operations *
  action            deny
 exit
 rule any-access
  action permit
 exit
exit
nacm rule-list confd-api-manager
 group [ confd-api-manager ]
 rule any-access
  action permit
 exit
exit
nacm rule-list ops-center-security
 group [ * ]
 rule change-self-password
  module-name       ops-center-security
  path              /smiuser/change-self-password
  access-operations exec
  action            permit
 exit
 rule smiuser
```

```
  module-name       ops-center-security
  path              /smiuser
  access-operations exec
  action            deny
 exit
exit
nacm rule-list lawful-intercept
 group [ LI ]
 rule li-accept-tap
  module-name       lawful-intercept
  path              /lawful-intercept
  access-operations *
  action            permit
 exit
 rule li-accept-clear
  module-name       tailf-mobile-smf
  path              /clear/lawful-intercept
  access-operations *
  action            permit
 exit
exit
nacm rule-list any-group
 group [ * ]
 rule li-deny-tap
  module-name       lawful-intercept
  path              /lawful-intercept
  access-operations *
  action            deny
 exit
 rule li-deny-clear
  module-name       tailf-mobile-smf
  path              /clear/lawful-intercept
  access-operations *
  action            deny
 exit
exit
```

Day 1 構成のロード

cnSGW-C の構成は、Ops センターのインフラストラクチャを使用して提供されます。Day 1 構成をロードするには、次のコマンドを実行します。

ssh admin*@ops_center_pod_ip* **-p 2024** *Day1config.cli*

（注） Day1config.cli （26 ページ） ファイルには、Day 1 構成に必要なパラメータが含まれています。

または、構成をコピーして cnSGW-C Ops センター CLI に貼り付け、Day 1 構成をロードすることもできます。

```
config
  <Paste the Day 1 configuration here>
  commit
  end
```

Day1config.cli

次に、cnSGW-C の Day 1 構成が含まれている Day1config.cli ファイルのサンプルを示します。

```
ipam
instance 1
source local
address-pool poolv4
vrf-name ISP
tags
dnn intershat
dnn starent.com
exit
ipv4
split-size
per-cache 1024
per-dp    256
exit
address-range 209.165.200 209.165.200.224
exit
exit
address-pool poolv4DNN2
vrf-name ISP
tags
dnn intershat1
exit
ipv4
split-size
per-cache 1024
per-dp    256
exit
address-range 209.165.100 209.165.201.0
exit
exit
address-pool poolv4DNN3
static
vrf-name ISP
tags
dnn intershat2
exit
ipv4
split-size
per-cache 512
per-dp    512
exit
address-range 209.165.202 209.165.202.128
exit
ipv6
prefix-ranges
split-size
per-cache 8192
per-dp    8192
exit
prefix-range 2002:db0:: length 48
exit
exit
exit
address-pool poolv4vDNN
vrf-name ISP
tags
dnn intershat1
exit
```

```
ipv4
split-size
per-cache 1024
per-dp    256
exit
address-range 209.165.200 209.165.202.128
exit
exit
address-pool poolv6
vrf-name ISP
tags
dnn intershat
exit
ipv6
prefix-ranges
split-size
per-cache 8192
per-dp    1024
exit
prefix-range 2001:db0:: length 48
exit
exit
exit
address-pool poolv6DNN2
vrf-name ISP
tags
dnn intershat1
exit
ipv6
prefix-ranges
split-size
per-cache 8192
per-dp    1024
exit
prefix-range 2001:ef0:: length 48
exit
exit
exit
address-pool poolv6vDNN
vrf-name ISP
tags
dnn intershat1
exit
ipv6
prefix-ranges
split-size
per-cache 8192
per-dp    1024
exit
prefix-range 2001:ab0:: length 48
exit
exit
exit
exit
cdl deployment-model small
cdl zookeeper replica 1
cdl datastore session
 slice-names 1
index map    1
index write-factor 1
slot replica 1
slot map     1
slot write-factor 1
exit
```

```
cdl kafka replica 1
etcd replicas 1
instances instance 1
 slice-name 1
 system-id  DCNAME001
 cluster-id CLUSTER0001
exit
local-instance instance 1
instance instance-id 1
endpoint sbi
replicas 1
vip-ip 209.165.201.3 vip-port 1234

  interface nrf
   loopbackPort 9001
   sla response 1000
   sla procedure 1000
   vip-ip 209.165.201.3 vip-port 9002 offline
  exit
  interface n11
   loopbackPort 9011
   sla response 1000
   sla procedure 1000
   vip-ip 209.165.201.3 vip-port 8090
  exit
  interface n7
   loopbackPort 9007
   sla response 1000
   sla procedure 1000
   vip-ip 209.165.201.3 vip-port 8090
  exit
  interface n10
   loopbackPort 9010
   sla response 1000
   sla procedure 1000
   vip-ip 209.165.201.3 vip-port 8090
  exit
  interface n40
   loopbackPort 9040
   sla response 1000
   sla procedure 1000
   vip-ip 209.165.201.3 vip-port 8090
  exit


exit
endpoint li
replicas 1
vip-ip 209.165.201.3
exit
endpoint nodemgr
replicas 1
nodes    1
exit
endpoint gtp
replicas 1
interface s5
vip-ip 209.165.200.225
exit
interface s2b
vip-ip 209.165.200.225
exit
interface s5e
vip-ip 209.165.201.3
```

```
exit
interface s11
vip-ip 209.165.200.226
exit
exit
endpoint pfcp
replicas 1
enable-cpu-optimization true
interface sxa
heartbeat
interval               5
retransmission-timeout 3
max-retransmissions    5
exit
interface n4
heartbeat
interval               0
retransmission-timeout 3
max-retransmissions    5
exit
exit
exit
#endpoint radius-dns
#replicas 1
#vip-ip 209.165.201.3
#interface radius-client
#vip-ip 209.165.201.3
#exit
#exit
endpoint service
replicas 1
nodes 1
exit
endpoint protocol
vip-ip 209.165.201.3
replicas 1
interface n4
vip-ip 209.165.200.225
exit
interface sxa
vip-ip 209.165.201.3
exit
exit
endpoint sgw-service
replicas 1
node 1
exit
exit
logging level application debug
logging level transaction debug
logging level tracing debug
logging name infra.config.core level application trace
logging name infra.config.core level transaction trace
logging name infra.config.core level tracing off
logging name infra.message_log.core level transaction trace
deployment
 model small
 app-name     SMF
 cluster-name Local
 dc-name      DC
exit
k8 label protocol-layer key disktype value ssd
#k8 label service-layer key radnaik_key value mine
#k8 label service-layer key smi.cisco.com/node-type value oam
```

```
exit
system mode running
helm default-repository cn
helm repository cn
#access-token smf-deployer.gen:Mitg_123
#access-token dev-deployer.gen:Mitg_123
#access-token
dev-deployer.gen:AKCp5ekcXA7TknM9DbLASNBw4jwVEsx9Z9WpQwEvCvCQ2mJhLymcz6BfbH38YJiWC6fn1cKmw
access-token
smf-deployer.gen:AKCp5ekcX7DcBhuAmMZYfGLaHvH3E4Syr9TQDp1gjzcSjYrqsrGbXSYs5X2XYij3d9n9VfWQe
#url
https://engci-maven-master.cisco.com/artifactory/smi-fuse-internal-snapshot/mobile-cnat-cn/cn-products/dev-cn-stage
url
https://engci-maven-master.cisco.com/artifactory/smi-fuse-internal-snapshot/mobile-cnat-cn/cn-products/dev-cn-stage
exit
profile nf-client nf-type udm
udm-profile UP1
locality LOC1
priority 30
service name type nudm-sdm
endpoint-profile EP1
capacity   30
uri-scheme http
version
uri-version v2
exit
exit
endpoint-name EP1
primary ip-address ipv4 209.165.201.3
primary ip-address port 8001
exit
exit
exit
service name type nudm-uecm
endpoint-profile EP1
capacity   30
uri-scheme http
endpoint-name EP1
primary ip-address ipv4 209.165.201.3
primary ip-address port 8001
exit
exit
exit
service name type nudm-ee
endpoint-profile EP1
capacity       30
api-uri-prefix PREFIX
api-root       ROOT
uri-scheme     http
endpoint-name EP1
priority 56
primary ip-address ipv4 209.165.201.3
primary ip-address port 8001
exit
exit
exit
exit
exit
exit
profile nf-client nf-type pcf
pcf-profile PP1
locality LOC1
priority 30
service name type npcf-am-policy-control
```

```
endpoint-profile EP1
capacity   30
uri-scheme http
endpoint-name EP1
priority 56
primary ip-address ipv4 209.165.201.3
primary ip-address port 8003
exit
exit
exit
service name type npcf-smpolicycontrol
endpoint-profile EP1
capacity   30
uri-scheme http
endpoint-name EP1
priority 56
primary ip-address ipv4 209.165.201.3
primary ip-address port 8003
exit
exit
exit
exit
exit
exit
profile nf-client nf-type amf
amf-profile AP1
locality LOC1
priority 30
service name type namf-comm
endpoint-profile EP2
capacity   30
uri-scheme http
endpoint-name EP1
priority 56
primary ip-address ipv4 209.165.201.3
primary ip-address port 8002
exit
exit
exit
exit
exit
exit
profile nf-client nf-type chf
chf-profile CP1
locality LOC1
priority 30
service name type nchf-convergedcharging
endpoint-profile EP1
capacity   30
uri-scheme http
version
uri-version v2
exit
exit
endpoint-name EP1
priority 56
primary ip-address ipv4 209.165.201.3
primary ip-address port 8004
exit
exit
exit
exit
exit
chf-profile CP2
```

```
locality LOC1
priority 31
service name type nchf-convergedcharging
endpoint-profile EP1
capacity   30
uri-scheme http
version
uri-version v2
exit
exit
endpoint-name EP1
priority 56
primary ip-address ipv4 209.165.201.3
primary ip-address port 9040
exit
exit
exit
exit
exit
exit
profile nf-pair nf-type UDM
nrf-discovery-group udmdiscovery
locality client  LOC1
locality preferred-server LOC1
locality geo-server GEO
exit
profile nf-pair nf-type AMF
nrf-discovery-group udmdiscovery
locality client  LOC1
locality preferred-server LOC1
locality geo-server GEO
exit
profile nf-pair nf-type PCF
nrf-discovery-group udmdiscovery
locality client  LOC1
locality preferred-server LOC1
locality geo-server GEO
exit
profile nf-pair nf-type UPF
nrf-discovery-group udmdiscovery
locality client  LOC1
locality preferred-server LOC1
locality geo-server GEO
exit
profile nf-pair nf-type CHF
nrf-discovery-group udmdiscovery
locality client  LOC1
locality preferred-server LOC1
locality geo-server GEO
exit
profile nf-client-failure nf-type udm
profile failure-handling FH4
service name type nudm-sdm
message type UdmSdmGetUESMSubscriptionData
status-code httpv2 403
retry  3
action retry-and-ignore
exit
status-code httpv2 404
action continue
exit
status-code httpv2 413
retry  3
action retry-and-continue
```

```
exit
status-code httpv2 501
retry  3
action retry-and-terminate
exit
status-code httpv2 503
action terminate
exit
status-code httpv2 504
retry  3
action retry-and-terminate
exit
exit
message type UdmSdmSubscribeToNotification
status-code httpv2 403
retry  3
action retry-and-ignore
exit
status-code httpv2 404
action continue
exit
status-code httpv2 413
retry  3
action retry-and-continue
exit
status-code httpv2 501
retry  3
action retry-and-terminate
exit
status-code httpv2 503
action terminate
exit
status-code httpv2 504
retry  3
action retry-and-terminate
exit
exit
exit
service name type nudm-uecm
message type UdmUecmRegisterSMF
status-code httpv2 403
retry  3
action retry-and-ignore
exit
status-code httpv2 404
action continue
exit
status-code httpv2 413
retry  3
action retry-and-continue
exit
status-code httpv2 501
retry  3
action retry-and-terminate
exit
status-code httpv2 503
action terminate
exit
status-code httpv2 504
retry  3
action retry-and-terminate
exit
exit
exit
```

```
exit
exit
profile nf-client-failure nf-type pcf
profile failure-handling FH1
service name type npcf-smpolicycontrol
message type PcfSmpolicycontrolCreate
status-code httpv2 0
action retry-and-ignore
exit
status-code httpv2 400
action continue
exit
status-code httpv2 403
action retry-and-ignore
exit
status-code httpv2 404
action terminate
exit
status-code httpv2 500
retry  2
action retry-and-ignore
exit
status-code httpv2 503
retry  2
action retry-and-continue
exit
exit
message type PcfSmpolicycontrolUpdate
status-code httpv2 0
action retry-and-ignore
exit
status-code httpv2 400
action continue
exit
status-code httpv2 403
action retry-and-ignore
exit
status-code httpv2 404
action terminate
exit
status-code httpv2 500
retry  2
action retry-and-ignore
exit
status-code httpv2 503
retry  2
action retry-and-continue
exit
exit
message type PcfSmpolicycontrolDelete
status-code httpv2 0
action retry-and-ignore
exit
status-code httpv2 400
action continue
exit
status-code httpv2 403
action retry-and-ignore
exit
status-code httpv2 404
action terminate
exit
status-code httpv2 500
retry  2
```

```
action retry-and-ignore
exit
status-code httpv2 503
retry  2
action retry-and-continue
exit
exit
exit
exit
exit
profile nf-client-failure nf-type chf
profile failure-handling FH2
service name type nchf-convergedcharging
message type ChfConvergedchargingCreate
status-code httpv2 0
action continue
exit
status-code httpv2 400
retry  3
action retry-and-terminate
exit
status-code httpv2 403
retry  3
action retry-and-ignore
exit
status-code httpv2 404
retry  3
action retry-and-terminate
exit
status-code httpv2 500
action continue
exit
status-code httpv2 503
action terminate
exit
status-code httpv2 504
action continue
exit
exit
message type ChfConvergedchargingUpdate
status-code httpv2 0
action continue
exit
status-code httpv2 400
retry  3
action retry-and-terminate
exit
status-code httpv2 403
retry  3
action retry-and-ignore
exit
status-code httpv2 404
retry  3
action retry-and-terminate
exit
status-code httpv2 500
action continue
exit
status-code httpv2 503
action terminate
exit
status-code httpv2 504
action continue
exit
```

```
exit
message type ChfConvergedchargingDelete
status-code httpv2 0
action continue
exit
status-code httpv2 400
retry  3
action retry-and-terminate
exit
status-code httpv2 403
retry  3
action retry-and-ignore
exit
status-code httpv2 404
retry  3
action retry-and-terminate
exit
status-code httpv2 500
action continue
exit
status-code httpv2 503
action terminate
exit
status-code httpv2 504
action continue
exit
exit
exit
exit
exit
profile sgw sgw1
 locality          LOC2
 fqdn              cisco.com.apn.epc.mnc456.mcc123
 #subscriber-policy polSub
exit
profile smf smf1
node-id       abcdef
locality      LOC1
fqdn          cisco.com.apn.epc.mnc456.mcc123
allowed-nssai [ slice1 ]
plmn-id mcc 123
plmn-id mnc 456
service name nsmf-pdu
type               pdu-session
schema             http
service-id         1
version            1.Rn.0.0
http-endpoint base-url http://smf-service
icmpv6-profile     icmpprf1
compliance-profile comp1
access-profile     access1
subscriber-policy  polSub
exit
exit
profile sgw sgw1
locality      LOC2
fqdn          cisco.com.apn.epc.mnc456.mcc123
plmn-id mcc 123
plmn-id mnc 456
#subscriber-policy  polSub
exit
profile dnn starent.com
 network-element-profiles chf chf1
 network-element-profiles amf amf1
```

```
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 charging-profile chgprf1
 virtual-mac      b6:6d:47:47:47:47
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV6 IPV4V6 ]
 upf apn starent.com
 #dcnr true
exit

profile dnn default-profile
 network-element-profiles chf chf1
 network-element-profiles amf amf1
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 charging-profile chgprf1
 virtual-mac      b6:6d:47:47:47:47
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV6 IPV4V6 ]
 upf apn starent.com
 #dcnr true
exit

profile dnn intershat
network-element-profiles chf chf1
network-element-profiles amf amf1
network-element-profiles pcf pcf1
network-element-profiles udm udm1
charging-profile chgprf1
virtual-mac      b6:6d:47:47:47:47
ssc-mode 2 allowed [ 3 ]
session type IPV4 allowed [ IPV6 IPV4V6 ]
upf apn intershat
dcnr             true
exit
profile dnn intershat1
network-element-profiles chf chf1
network-element-profiles amf amf1
network-element-profiles pcf pcf1
network-element-profiles udm udm1
charging-profile chgprf1
virtual-mac      b6:6d:47:47:47:48
pcscf-profile    PCSCF_Prof_2
ssc-mode 1
session type IPV4
exit
profile dnn intershat2
network-element-profiles chf chf
network-element-profiles amf amf
network-element-profiles pcf pcf
network-element-profiles udm udm
charging-profile chgprf1
virtual-mac      b6:6d:47:47:47:49
ssc-mode 2 allowed [ 3 ]
session type IPV4 allowed [ IPV6 IPV4V6 ]
upf apn intershat2
exit
profile qos abc
ambr ul "250 Kbps"
ambr dl "500 Kbps"
qi5      7
arp priority-level 14
arp preempt-cap NOT_PREEMPT
arp preempt-vuln PREEMPTABLE
```

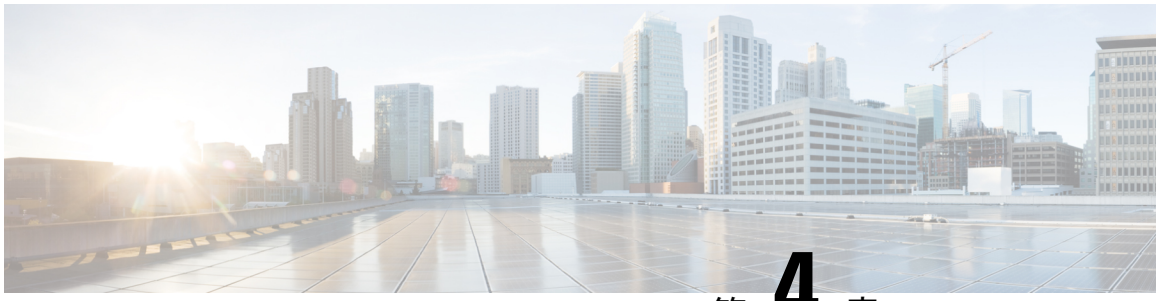
```
priority 120
max data-burst 2000
exit
profile failure-handling FH1
interface pfcp message N4SessionEstablishmentReq
cause-code pfcp-entity-in-congestion action retry-terminate max-retry 2
cause-code system-failure action terminate
cause-code service-not-supported action terminate
cause-code no-resource-available action retry-terminate max-retry 3
cause-code no-response-received action retry-terminate max-retry 1
cause-code reject action terminate
exit
interface pfcp message N4SessionModificationReq
cause-code mandatory-ie-incorrect action terminate
cause-code session-ctx-not-found action terminate
cause-code reject action terminate
exit
exit
profile failure-handling gtp1
interface gtpc message S5S8CreateBearerReq
cause-code temp-fail
action retry timeout 1000 max-retry 2
exit
exit
interface gtpc message S5S8UpdateBearerReq
cause-code temp-fail
action retry timeout 1000 max-retry 2
exit
exit
interface gtpc message S5S8DeleteBearerReq
cause-code temp-fail
action retry timeout 1000 max-retry 2
exit
exit
exit
profile network-element amf amf1
nf-client-profile        AP1
failure-handling-profile FH3
query-params [ dnn ]
exit
profile network-element pcf pcf1
nf-client-profile        PP1
failure-handling-profile FH1
query-params [ dnn ]
rulebase-prefix          cbn#
predefined-rule-prefix   crn#
exit
profile network-element udm udm1
nf-client-profile        UP1
failure-handling-profile FH4
query-params [ dnn ]
exit
profile network-element upf upf226
node-id upf226@sgw.com
n4-peer-address ipv4 209.165.201.4
n4-peer-port 8805
dnn-list     [ intershat intershat1 intershat2 cisco.com starent.com ]
capacity     2000
priority     10
exit
profile network-element upf upf1
node-id upf1@sgw.com
n4-peer-address ipv4 209.165.201.5
n4-peer-port 8805
```

```
dnn-list     [ intershat intershat1 intershat2 cisco.com starent.com ]
capacity     2000
priority     10
exit
profile network-element upf upf2
node-id upf2@sgw.com
n4-peer-address ipv4 209.165.201.6
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 cisco.com starent.com ]
capacity     2000
priority     1
exit
profile network-element upf upf76
node-id upf3@sgw.com
n4-peer-address ipv4 209.165.201.7
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     1000
priority     10
exit
profile network-element upf upf70
node-id upf4@sgw.com
n4-peer-address ipv4 209.165.201.8
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     1000
priority     10
exit
profile network-element upf upf71
node-id upf5@sgw.com
n4-peer-address ipv4 209.165.201.9
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     1000
priority     10
exit
profile network-element upf upf72
n4-peer-address ipv4 209.165.201.10
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
profile network-element upf upf79
n4-peer-address ipv4 209.165.201.11
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
profile network-element upf upf131
n4-peer-address ipv4 209.165.201.12
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
profile network-element upf upf132
n4-peer-address ipv4 209.165.201.13
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
```

```
profile network-element upf upf133
n4-peer-address ipv4 209.165.201.14
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
profile network-element upf upf134
n4-peer-address ipv4 209.165.201.15
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
profile network-element upf upf135
n4-peer-address ipv4 209.165.201.16
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
profile network-element upf upf136
n4-peer-address ipv4 209.165.201.17
n4-peer-port 8805
dnn-list     [ intershat1 intershat2 starent.com cisco.com ]
capacity     2000
priority     10
exit
profile network-element chf chf1
nf-client-profile         CP1
failure-handling-profile  FH2
query-params [ dnn ]
nf-client-profile-offline CP2
exit
profile network-element chf chgser1
exit
profile compliance comp1
service nsmf-pdusession
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service namf-comm
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service n1
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service n2
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service nudm-sdm
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service nudm-uecm
version uri v1
```

```
version full 1.0.0
version spec 15.4.0
exit
service nnrf-disc
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service nnrf-nfm
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service npcf-smpolicycontrol
version uri v1
version full 1.0.0
version spec 15.4.0
exit
service nchf-convergedcharging
version uri v1
version full 1.0.0
version spec 15.3.0
exit
exit
profile upf-group group1
failure-profile FH1
exit
profile access access1
n26 idft enable timeout 15
n2 idft enable timeout 15
gtpc gtpc-failure-profile gtp1
exit
profile icmpv6 icmpprf1
options virtual-mac b6:6d:57:45:45:45
exit
profile charging chgprf1
method [ offline ]
exit
profile charging-characteristics 1
charging-profile chgprf1
exit
nssai name slice1
sst 2
sdt Abf123
dnn [ dnn1 intershat intershat1 intershat2 ]
exit
policy subscriber polSub
precedence 1
sst              02
sdt              Abf123
serving-plmn mcc 123
serving-plmn mnc 456
supi-start-range 100000000000001
supi-stop-range  999999999999999
gpsi-start-range 1000000000
gpsi-stop-range  9999999999
operator-policy  opPol1
exit
precedence 511
operator-policy defOprPol1
exit
exit
policy operator defOprPol1
policy dnn         defPolDnn
```

```
policy network-capability nc1
exit
policy operator opPol1
policy dnn         polDnn
policy network-capability nc1
exit
policy dnn defPolDnn
profile default-profile
dnn dnn2 profile profile2
dnn intershat profile intershat
dnn intershat1 profile intershat1
dnn starent.com profile starent.com
exit
policy dnn polDnn
profile default-profile
dnn dnn2 profile profile2
dnn intershat profile intershat
dnn intershat1 profile intershat1
dnn intershat2 profile intershat2
dnn starent.com profile starent.com
exit
policy network-capability nc1
nw-support-local-address-tft true
exit
nacm groups group LI2
user-name [ liadmin2 ]
exit
nacm groups group LI3
user-name [ liadmin3 ]
exit
nacm groups group admin
user-name [ admin ]
exit
commit
end
```



第 4 章

スマートライセンスサポート

- 機能の概要と変更履歴 （43 ページ）
- スマート ソフトウェア ライセンシング （44 ページ）
- スマート ライセンスの設定 （51 ページ）
- スマートライセンス情報の表示 （59 ページ）

機能の概要と変更履歴

要約データ

表 *3* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	無効：設定が必要
関連資料	該当なし

マニュアルの変更履歴

表 *4:* マニュアルの変更履歴

改訂の詳細	リリース
拡張機能が導入されました。 複数の利用資格タグ：cnSGW-C は、インストールされているサービスプロファイルに基づいてソフトウェアライセンスの利用資格情報を返す REST サービスをサポートしています。	2021.02.0
最初の導入。	2020.03.0

スマート ソフトウェア ライセンシング

シスコでは、従来のライセンシングとスマート ソフトウェア ライセンシングの 2 種類のライセンスモデルを採用しています。従来のライセンシングは、製品アクティベーションキー（PAK） をシスコ製品にインストールすることによって、ソフトウェア アクティベーションで構成されます。製品アクティベーションキーは、他のシスコ機器と同様の方法で発注し、シスコ製品の機能セットのライセンスファイルを取得するために使用する購入可能品目です。この従来のライセンスの場合、シスコライセンスサーバーとのオンライン通信は不要です。

スマート ソフトウェア ライセンシングは、ライセンスレポートを承認および提供するいくつかのツールを使用する、エンドツーエンドプラットフォームのクラウドベースのライセンシングです。NF に組み込まれたスマート ソフトウェア ライセンシング機能により、製品の登録と承認が完了します。cnSGW-C は、スマート ソフトウェア ライセンシング モデルをサポートしています。

スマートライセンシングは、シスコのソフトウェア資産の購入、展開、管理を簡素化します。権限付与は Cisco Commerce Workspace（CCW）を介し、シスコアカウントを通じて購入され、使用するためにバーチャルアカウントにすぐに取り込まれます。このアプローチにより、すべてのデバイスにライセンスファイルをインストールする必要がなくなります。スマート対応製品では、レポートの使用状況がシスコに直接通知されます。Cisco Software Central では、お客様は Cisco ソフトウェアライセンスを1 つの場所で管理できます。ライセンスの所有権と使用状況に関する情報をすぐに利用でき、使用状況やビジネスニーズに基づいてより良い購入判断ができます。

Cisco Smart Licensing の詳細については、
https://www.cisco.com/c/en/us/buy/smart-accounts/software-licensing.html を参照してください。

Cisco Software Central

Cisco Software Central（CSC）を使用すると、ソフトウェアライセンスとスマートアカウントを単一のポータルから管理できます。CSCインターフェイスを使用して、製品の有効化、権限付与の管理、更新、ソフトウェアのアップグレードを行うことができます。登録プロセスを完了するには、機能しているスマートアカウントが必要です。

Cisco Software Central にアクセスするには、https://software.cisco.com を参照してください。

スマート アカウントとバーチャル アカウント

スマートアカウントでは、スマート対応のすべての製品と権限付与を 1 つの場所で管理します。これにより、シスコソフトウェアの調達、展開、およびメンテナンスを迅速に行うことができます。スマートアカウントを作成するには、要求元の組織を代表する権限が必要です。送信後、要求は承認プロセスを経由します。

バーチャルアカウントは、スマートアカウント内のサブアカウントとして存在します。バーチャルアカウントは、組織の配置、ビジネス機能、地域、または定義された階層に基づいてお客様が定義します。スマートアカウント管理者は、バーチャルアカウントを作成および維持できます。

スマートアカウントのセットアップまたは管理の詳細については、https://software.cisco.com を参照してください。

Cisco スマートアカウントの要求

Cisco スマートアカウントとは、スマートライセンス対応製品が利用可能なアカウントです。Cisco スマートアカウントでは、デバイスのライセンスの管理とアクティブ化、ライセンス使用状況のモニター、シスコライセンスの購入の追跡が行えます。透過的なアクセスにより、スマートライセンス製品をリアルタイムで表示できます。IT 管理者は、Cisco Software Central を使用して、組織のスマートアカウント内のライセンスとアカウントユーザーを管理できます。Cisco スマートアカウントを作成するには、次の手順を実行します。

手順

ステップ 1 次の URL にアクセスしてください。

```
https://software.cisco.com
```

ステップ 2 クレデンシャルを使用してログインし、[Administration] 領域で [Request a Smart Account] をクリックします。

[Smart Account Request] ウィンドウが表示されます。

ステップ 3 [Create Account] で、次のいずれかのオプションを選択します。

- [Yes, I have authority to represent my company and want to create the Smart Account]。このオプションを選択した場合は、組織を代表して製品とサービスの利用資格、ユーザー、およびロールを作成し、管理する権限に同意したことになります。
- [No, the person specified below will create the account]。このオプションを選択した場合は、スマートアカウントを作成する担当者の電子メールアドレスを入力する必要があります。

ステップ 4 [Account Information] で、

a) [Account Domain Identifier] の横にある [Edit] をクリックします。
b) [Edit Account Identifier] ダイアログボックスで、ドメインを入力し、[OK] をクリックします。デフォルトでは、ドメインはアカウントを作成する担当者の電子メールアドレスに基づいており、このアカウントを所有する企業に帰属している必要があります。
c) [Account Name] に入力します（通常は会社名）。

ステップ 5 [Continue] をクリックします。
スマートアカウント要求は、アカウントドメイン識別子によって承認されるまで保留中の状態になります。承認後、設定プロセスを実行するための手順が記載された確認メールが送信されます。

cnSGW-C スマートライセンス

スマートライセンス機能は、すべての 5G アプリケーションのオンラインライセンスとオフラインライセンスのアプリケーション権限をサポートしています。アプリケーションの使用は、コンプライアンス違反（OOC）と期限切れの段階を含む、ライセンスのすべての段階で制限されません。

（注） 使用中のすべてのライセンスに対して 90 日間の評価期間が付与されます。現在、評価期間の終了後も、5Gアプリケーションの機能と操作は制限されません。

ソフトウェアタグと権限付与タグ

次のセクションでは、ライセンスの識別、レポート、および適用のために作成されるソフトウェアタグと利用資格タグについての情報を提供します。

ソフトウェアタグ

ソフトウェアタグまたは製品タグは、スマートライセンスシステムがソフトウェア製品ファミリを識別するのに役立つ一意の識別子です。Cisco Smart Software Manager でスマート製品インスタンスを追加する際、スマートクライアントはソフトウェア/製品タグを識別のために使用します。

cnSGW-C には、次のソフトウェアタグがあります。

製品タイプの説明	ソフトウェアタグ
Ultra Cloud Core：サービングゲートウェイ機能（cnSGWc）、基本最小値	regid.2020-07.com.cisco.cnSGWc,1.0_ff0b64f1-f54d-46d3-afc7-052d41870b59

利用資格タグ

利用資格タグはソフトウェアの一部であり、ソフトウェアイメージで使用されている機能を識別します。これらのタグは、デバイス上にインストールされているソフトウェア製品の使用状況と権限付与に関する通信の基盤となります。利用資格タグは、PID ライセンスとソフトウェアイメージの両方にマッピングされます。すべてのスマート対応 PID に、1 つ以上の利用資格タグが含まれている場合があります。

次の利用資格タグで、使用中のライセンスを識別します。

製品タイプの説明	権限付与タグ
Ultra Cloud Core：サービングゲートウェイ機能（cnSGWc）、1K セッション	regid.2020-07.com.cisco.cnSGWc_1K,1.0_6ce36c73-26dd-4607-ab9b-077fbb2e0f54

（注） ソフトウェアのアップグレードとロールバックの間、ライセンス情報は保持されます。

複数の権限付与タグ

機能説明

cnSGW-C は、REST エンドポイント構成をサポートしています。この REST エンドポイントは、インストールされているサービスプロファイルに基づいてソフトウェアライセンスの権限付与情報を返す REST サービスをサポートしています。次に例を示します。

- スタンドアロン SMF
- スタンドアロン cnSGW-C
- SMF と cnSGW-C の組み合わせ

（注） この機能は、コンバージドコア製品にのみ適用されます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

複数の権限付与タグを設定するには、内部ポート 8000 の NF（cnSGW-C/SMF）REST-EP ポッドに追加された GET サービスを使用します。REST サービス名は「entitlements」です。

（注） `localhost:8000` はすでに権限付与サービスによって占有されているため、ポート 8000 および REST-EP 内の localhost で新しいサービスを作成しないことを推奨します。

Ops Center の `values.yaml` は、このサービスを製品構成の一部として登録します。

以下に設定例を示します。

```
ops-center:
  product:
    id: <product_id>, e.g. SMF
    softwareID: <s/w id>, e.g.
regid.2020-04.com.cisco.SMF,1.0_37ffdc21-3e95-4192-bcda-d3225b6590ce
    entitlementsURL: http://entitlements:8000/entitlements.json
```

`values.yaml` に entitlementsURL が入力されると、Ops Center は、権限付与サービスから受け取った使用可能なすべてのライセンスをインストールします。

権限付与サービスは、`rest-ep-entitlements-cm` 設定マップで権限を検索し、使用可能なすべての権限を JSON 応答として返します。

`rest-ep-entitlements-cm` の権限は、次のフラグに基づいて登録されます。

- restep.smfProfile
- restep.sgwProfile

（注） フラグは、**cn-ops-center** > **confd_init** > **render** > **rest-ep** > **pod.yaml** で設定されています。

権限付与サービスに権限情報がない場合、Ops Center はスマートライセンスサーバーに要求を送信しないか、ライセンスをインストールしません。

SNMP トラップ

製品が契約に準拠していない場合（製品で使用しているライセンスや権限が多すぎるか、特定の権限付与タグの使用が許可されていない場合）、権限付与タグを使用してすべてのアプリケーションに通知が送信されます。準拠していない権限を示す SNMP トラップが送信されます。この SNMP トラップは、トラップ名が `SMART_LIC-3-OUT_OF_COMPLIANCE` であるスマートエージェントの syslog に表示されます。

制限事項

コンバージドコアには、SMF と cnSGW-C の 2 つのサービスプロファイルがあり、各サービスには固有の製品 ID が割り当てられています。ソフトウェア ライセンス サーバーに登録する際に、SMF と cnSGW-C はそれぞれの製品 ID とその権限を送信します。

スマートエージェントは、複数の製品 ID の処理をサポートしていません。スマートエージェントによる処理には SMF 製品 ID を使用することをお勧めします。

設定例

次に、`rest-ep-entitlements-cm` configmap の設定例を示します。

```
Name:         rest-ep-entitlements-cm
Namespace:    smf
Labels:       app=rest-ep
              app.kubernetes.io/managed-by=Helm
              chart=rest-ep-0.5.2-dev-multi-entitlement-7600-210225084534-a5b5b67
              component=rest-ep
              heritage=Helm
              release=smf-rest-ep
Annotations:  meta.helm.sh/release-name: smf-rest-ep
              meta.helm.sh/release-namespace: smf

Data
====
nf-profiles:
----
configuredProfiles:
- name: "smf"
  entitlement:
    displayName: "UCC 5G SMF BASE"
    entitlementTag:
regid.2020-04.com.cisco.SMF_BASE,1.0_b49f5997-21aa-4d15-9606-0cff88729f69
    entitlementVersion: "1.0"
- name: "sgw"
  entitlement:
    displayName: "UCC cnSGWc 1K"
    entitlementTag:
regid.2020-07.com.cisco.cnSGWc_1K,1.0_6ce36c73-26dd-4607-ab9b-077fbb2e0f54
    entitlementVersion: "1.0"
Events:  <none>

JSON response format from REST API http://entitlements:8000/entitlemens.json

[
{
  "displayName" : "SMF_BASE",
  "entitlementTag" : "
regid.2020-04.com.cisco.SMF_BASE,1.0_b49f5997-21aa-4d15-9606-0cff88729f69",
  "entitlementVersion" : "1.0"

},
{
  "displayName" : "cnSGW_BASE",
  "entitlementTag" :
"regid.2020-02.com.cisco.cnSGW_BASE,1.0_a61f0740-ef15-4ac2-916f-77257902b22",
  "entitlementVersion" : "1.0"

}
]
```

設定チェック

ここでは、設定チェックについて説明します。

- 複数の利用資格タグを設定した後で、次のチェックを行う必要があります。

- NF の Ops センターが正常に展開されたことを確認します。
- 新しい展開と設定の後、すべてのポッドが稼働しており、準備完了状態であることを確認します（主に、サービス、nodemgr、cachepod、udp-proxy、rest-ep、およびプロトコルポッド）。
- SMF サービスがプロファイルで設定されている場合は、`rest-ep-entitlements-cm` に SMF 権限が入力されている必要があります。
- cnSGW-C サービスがプロファイルで設定されている場合は、`rest-ep-entitlements-cm` に cnSGW-C 権限が入力されている必要があります。
- SMF および cnSGW-C サービスの両方がプロファイルで設定されている場合、`rest-ep-entitlements-cm` には SMF および cnSGW-C の権限が入力されている必要があります。

- 複数の権限付与タグの設定を削除した後で、次のチェックを行う必要があります。
  - すべてのポッドが終了し、削除されていることを確認します（また、SMF は NRF から登録解除されます）。
  - すべてのセキュリティ関連アイテム（Ops センターで使用されるセキュリティアイテムを除く）が削除されていることを確認します。

トラブルシューティング

ここでは、トラブルシューティング情報について説明します。

- 権限付与サービスをトラブルシューティングするには、rest-ep ポッドのログを確認します。

  kubectl logs rest-ep-n0-0 -n *<namespace>* **-f**
- 権限付与サービスの問題をデバッグするために、次のコマンドの出力データを確認することもできます。
  - **show license tech-support**
  - **show license status**
  - **show license summary**
- スマートエージェントおよび Ops Center ポッドのトラブルシューティングには、次のコマンドを使用できます。
  - **kubectl logs** *<smart_agent_pod>* **-n namespace**
  - **kubectl logs** *<ops_center_pod>* **-n namespace**

スマート ライセンスの設定

新しい cnSGW-C の展開後にスマートライセンスを設定できます。

Cisco Software Central にアクセスできるユーザー

このセクションでは、お使いの環境から Cisco Software Central（CSC）ポータルにアクセスできる場合のスマートライセンスの設定方法について説明します。

CSC での製品と利用資格の設定

CSC で製品と利用資格を設定するには、次の手順を実行します。

1. CSC アカウントにログインします。
2. [Add Product] をクリックして、次の詳細を入力します。
   - [Product name]：展開された製品の名前を指定します。例：SGW。
   - [Primary PM CEC ID]：展開された製品のプライマリ プロジェクト マネージャの CEC ID を指定します。
   - [Dev Manager CEC ID]：展開された製品の開発マネージャの CEC ID を指定します。
   - [Description]：（任意）展開された製品の簡単な説明を指定します。
   - [Product Type]：製品タイプを指定します。
   - [Software ID Tag]：シスコアカウントチームから提供されたソフトウェア ID タグを指定します。
3. [作成（Create）] をクリックします。
4. [Product/Entitlement Setup] グリッドから製品を選択します。
5. [Entitlement] ドロップダウンをクリックし、[Create New Entitlement] を選択します。
6. [Add Entitlement] で [New Entitlement] を選択し、次の詳細を入力します。
   - [Entitlement Name]：ライセンス利用資格名を指定します。例：SGW_BASE。
   - [Description]：（任意）ライセンス権限付与に関する簡単な説明を指定します。
   - [Entitlement Tag]：シスコアカウントチームから提供された利用資格タグを指定します。
   - [Entitlement Type]：ライセンス利用資格のタイプを指定します。
   - [Vendor String]：ベンダー名を指定します。
7. [Entitlement Allocation] をクリックします。

8. [Add Entitlement Allocation] をクリックします。
9. [New License Allocation] で、次の詳細を入力します。
   - [Product]：ドロップダウンリストから製品を選択します。
   - [Entitlement]：ドロップダウンリストから利用資格を選択します。
10. [Continue] をクリックします。
11. [New License Allocation] で、次の詳細を入力します。
   - [Quantity]：ライセンスの数を指定します。
   - [License Type]：ライセンスのタイプを指定します。
   - [Expiring Date]：購入したライセンスの有効期限を指定します。
12. [作成（Create）] をクリックします。

スマートライセンスの登録

ライセンスの対象となる製品を CSC に登録する必要があります。製品を登録するには、CSC から ID トークンを生成する必要があります。

1. CSC アカウントにログインします。
2. [General] > [New Token] の順にクリックし、次の詳細を入力します。
   - [Description]：ID トークンの簡単な説明を指定します。
   - [Expires After]：トークンが期限切れになるまでの日数を指定します。
   - [Max. Number Users]：ユーザーの最大数を指定します。
3. [Create Token] をクリックします。
4. [Product Instance Registration Token] で新しい ID トークンを選択します。
5. [Actions] > [Copy] の順にクリックします。
6. cnSGW-C Ops センター CLI にログインし、次のコマンドを使用して ID トークン を貼り付けます。

   license smart register *idtoken*

   注：

   - **license smart register** ：CSC にスマートライセンスを登録します。
   - *idtoken* ：CSC から生成された ID トークンを指定します。

   例：

```
license smart register
Value for 'idtoken' (<string>):
```

```
MTI2Y2FlNTAtOThkMi00YTAxLWE4M2QtOTNhNzNjNjY4ZmFiLTE2MTc4N
Tky%0AMTA5MDh8ck1jUHNwc3k1ZC9nWFFCSnVEcUp4QU1jTFoxOGxDTU5kQ3lpa25E%0Ab04wST0%3D%0A
```

7. 次のコマンドを使用して、スマートライセンスのステータスを確認します。

   show license all

   例：

```
show license all

Smart Licensing Status
======================
Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Smart Account: CN-5G-NF
  Virtual Account: Default
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Jul 12 19:46:04 2020 GMT
  Last Renewal Attempt: SUCCEEDED on Jul 12 19:46:04 2020 GMT
  Next Renewal Attempt: Jan 8 19:46:04 2021 GMT
  Registration Expires: Jul 12 19:39:10 2021 GMT

License Authorization:
  Status: AUTHORIZED on Jul 12 19:46:06 2020 GMT
  Last Communication Attempt: SUCCEEDED on Jul 12 19:46:06 2020 GMT
  Next Communication Attempt: Aug 11 19:46:06 2020 GMT
  Communication Deadline: Oct 10 19:43:32 2020 GMT

License Conversion:
 Automatic Conversion Enabled: true
 Status: NOT STARTED

Utility:
  Status: DISABLED

Transport:
   Type: Smart Transport
   Registration URL: null
   Utility URL: null

Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 89 days, 1 hr, 20 min, 55 sec

License Usage
=============
License Authorization Status: AUTHORIZED as of Jul 12 19:46:06 2020 GMT

cnSGWc_1K (cnSGWc_1K)
  Description: Ultra Cloud Core - Serving Gateway Function (cnSGWc), 1K Sessions
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:cnSGWc,SN:JC5LXHI-2KVPPIQ

Agent Version
```

```
=============
Smart Agent for Licensing: 3.0.13
```

スマートライセンスの登録解除

スマートライセンスを登録解除するには、次の手順を実行します。

1. cnSGW-C Ops センター CLI にログインして、次のコマンドを使用します。

   license smart deregister

   注：

   - **license smart deregister** ：CSC からスマートライセンスを登録解除します。

2. 次のコマンドを使用して、スマートライセンスのステータスを確認します。

   show license all

   例：

```
show license all

Smart Licensing Status
======================
Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 89 days, 1 hr, 18 min, 55 sec
  Last Communication Attempt: NONE

License Conversion:
 Automatic Conversion Enabled: true
 Status: NOT STARTED

Utility:
  Status: DISABLED
Transport:
   Type: Smart Transport
   Registration URL: null
   Utility URL: null

Evaluation Period:
  Evaluation Mode: In Use
  Evaluation Period Remaining: 89 days, 1 hr, 18 min, 55 sec

License Usage
=============
License Authorization Status: EVALUATION MODE
  Evaluation Period Remaining: 89 days, 1 hr, 18 min, 55 sec

cnSGWc_1K (cnSGWc_1K)
  Description: Ultra Cloud Core - Serving Gateway Function (cnSGWc), 1K Sessions
  Count: 1
  Version: 1.0
  Status: EVAL MODE
  Export status: NOT RESTRICTED
```

```
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
====================
UDI: PID:cnSGWc,SN:JC5LXHI-2KVPPIQ

Agent Version
=============
Smart Agent for Licensing: 3.0.13
```

Cisco Software Central にアクセスできないユーザー

スマートライセンスの予約機能（永続予約）は、内部環境から CSC にアクセスできないお客様向けに予約されています。シスコでは、お客様が仮想カウントからライセンスを予約し、それをデバイスの固有デバイス識別子（UDI）に関連付けることができます。これにより、お客様は、予約済みライセンスを使用してデバイスを切断モードで使用できます。

次のセクションでは、内部環境から CSC にアクセスできないユーザーに向けてスマートライセンスを予約する手順について説明します。

スマートライセンス予約要求コードの有効化と生成

スマートライセンス予約要求コードを有効にし、生成するには、次の手順を実行します。

1. cnSGW-C Ops Center CLI にログインします。
2. 予約を有効にするには、次の構成を使用します。

   ```
   config terminal
     license smart reservation
     end
   ```

   注：

   - **license smart reservation** ：ライセンス予約を有効にします。

3. 予約コードを要求するには、次のコマンドを使用します。

   ```
   license smart reservation request
   ```

   注：

   - **license smart reservation request** ：予約要求コードを生成します。

重要 今後の使用のために、SGW Ops Center の CLI で生成されたライセンス要求コードをローカルマシンにコピーしてください。

例：

```
license smart reservation request
reservation-request-code CE-ZcnSGWc:JC5LXHI-2KVPPIQ-AwjEHYoEo-F8
```

```
Message from confd-api-manager at 2020-07-13 08:27:27...
Global license change NotifyReservationInProgress reason code Success - Successful.
```

CSC からの承認コードの生成

ライセンス予約要求コードを使用してCSCから承認コードを生成するには、次の手順を実行します。

1. CSC アカウントにログインします。
2. [ライセンスの予約（License Reservation）] をクリックします。
3. 要求コードの入力：SGW Ops Center の CLI からコピーしたライセンス予約要求コードを [Reservation Request Code] テキストボックスに貼り付けます。
4. ライセンスの選択：[Reserve a Specific License] オプションボタンをクリックし、[UCC 5G SGW BASE] を選択します。

（注） [Reserve] テキストボックスに、値 *1*を入力します。

5. 選択内容を確認します。
6. [承認コードを生成（Generate Authorization Code）] をクリックします。
7. 応答ファイルのダウンロード：承認コードが生成され、画面に表示されます。[Download as File] をクリックして、承認コードをダウンロードします。
8. [閉じる（**Close**）] をクリックします。

スマートライセンスの予約

CSC で生成された承認コードを使用して、展開された製品のスマートライセンスを予約するには、次の手順を実行します。

1. cnSGW-C Ops Center CLI にログインして、次のコマンドを使用します。

   license smart reservation install *authorization_code*

   注：

   - **license smart reservation install** *authorization_code* ：スマートライセンス承認コードをインストールします。

   例：

   ```
   license smart reservation install
   Value for 'key' (<string>): CAAAsJ-iwTYvW-puASse-nLGbcj-NJwnCo-EpxZ
   Message from confd-api-manager at 2020-07-13 08:30:00...
   Global license change NotifyReservationInstalled reason code Success - Successful.
   Message from confd-api-manager at 2020-07-13 08:30:01...
   Global license change NotifyRegisterSuccess reason code Success - Successful
   ```

2. 次のコマンドを使用して、スマートライセンスのステータスを確認します。

show license all

例：

```
show license all

Smart Licensing Status
======================
Smart Licensing is ENABLED
License Reservation is ENABLED

Registration:
  Status: REGISTERED - UNIVERSAL LICENSE RESERVATION
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Mon Jul 13 08:29:59 GMT 2020
  Last Renewal Attempt: None

License Authorization:
  Status: AUTHORIZED - RESERVED on Mon Jul 13 08:29:59 GMT 2020

Utility:
  Status: DISABLED

Transport:
   Type: Smart Transport
   Registration URL: null
   Utility URL: null

Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 88 days, 23 hr, 28 min, 54 sec

License Usage
=============
License Authorization Status:
  Status: AUTHORIZED - RESERVED on Mon Jul 13 08:29:59 GMT 2020
  Last Communication Attempt: SUCCEEDED on Jul 13 08:29:59 2020 GMT
  Next Communication Attempt: NONE
  Communication Deadline: NONE

cnSGWc_1K (cnSGWc_1K)
  Description: Ultra Cloud Core - Serving Gateway Function (cnSGWc), 1K Sessions
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: RESTRICTED_ALLOWED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:cnSGWc,SN:JC5LXHI-2KVPPIQ

Agent Version
=============
Smart Agent for Licensing: 3.0.13
```

予約済みライセンスの返却

予約済みのライセンスを返却するには、次の手順を実行します。

1. ライセンス予約承認コードが SGW Ops Center にインストールされている場合。

1. cnSGW-C Ops Center CLI にログインして、次のコマンドを使用します。

   license smart reservation return

   注：

   - **license smart reservation return**：予約済みのスマートライセンスを返却します。

   例：

   ```
   license smart reservation return
   reservation-return-code CAAsJA-vNGQbQ-YmwMTz-ZnN4Kb-eekEy7-jeo
   Message from confd-api-manager at 2020-07-13 08:32:37...
   Global license change NotifyReservationReturned reason code Success - Successful.
   ```

2. 今後の使用のために、SGW Ops Center の CLI で生成されたライセンス予約リターンコードをローカルマシンにコピーしてください。
3. CSC アカウントにログインします。
4. リストから製品インスタンスを選択します。
5. [Actions] > [Remove] をクリックします。
6. [Reservation Return Code] テキストボックスにリターンコードを貼り付けます。

2. ライセンス予約承認コードが SGW Ops Center にインストールされていない場合。
   1. cnSGW-C Ops Center CLI にログインし、次のコマンドを使用してリターンコードを生成します。

      license smart reservation return
      authorization_code

重要 CSC で生成されたライセンス予約承認コードを貼り付けて、リターンコードを生成します。

   2. CSC アカウントにログインします。
   3. リストから製品インスタンスを選択します。
   4. [Actions] > [Remove] をクリックします。
   5. [Reservation Return Code] テキストボックスにリターンコードを貼り付けます。

3. 次のコマンドを使用して、スマートライセンスのステータスを確認します。

   show license all

   例：

   ```
   show license all

   Smart Licensing Status
   ======================
   Smart Licensing is ENABLED
   ```

```
License Reservation is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 88 days, 23 hr, 23 min, 54 sec
  Last Communication Attempt: SUCCEEDED on Jul 13 08:29:59 2020 GMT
  Next Communication Attempt: NONE
  Communication Deadline: NONE

License Conversion:
 Automatic Conversion Enabled: true
 Status: NOT STARTED

Utility:
  Status: DISABLED

Transport:
   Type: Smart Transport
   Registration URL: null
   Utility URL: null

Evaluation Period:
  Evaluation Mode: In Use
  Evaluation Period Remaining: 88 days, 23 hr, 23 min, 54 sec

License Usage
=============
License Authorization Status: EVALUATION MODE
  Evaluation Period Remaining: 88 days, 23 hr, 23 min, 54 sec

cnSGWc_1K (cnSGWc_1K)
  Description: Ultra Cloud Core - Serving Gateway Function (cnSGWc), 1K Sessions
  Count: 1
  Version: 1.0
  Status: EVAL MODE
  Export status: RESTRICTED_NOTALLOWED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:cnSGWc,SN:JC5LXHI-2KVPPIQ

Agent Version
=============
Smart Agent for Licensing: 3.0.13
```

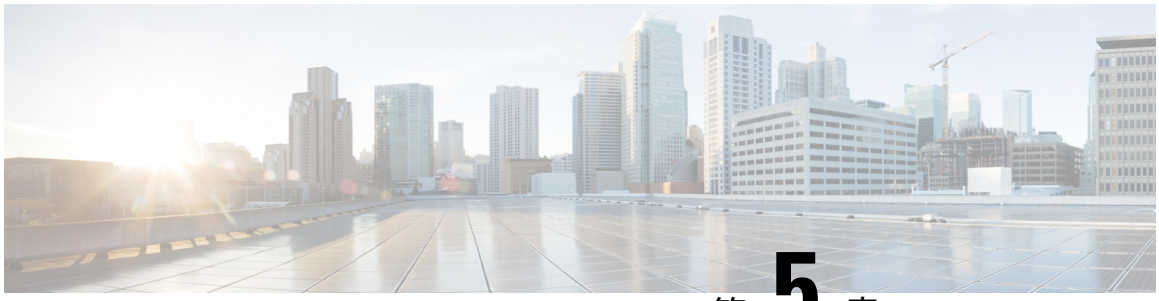
スマートライセンス情報の表示

次の **show license** コマンドを使用して、cnSGW-C Ops センターでスマートライセンス情報を表示します。

show license [all | UDI | displaylevel | reservation | smart | status | summary | tech-support | usage]

注：

- **all**：ライセンスステータス、使用状況、製品情報、スマート エージェント バージョンなどのスマートライセンス情報の概要を表示します。
- **UDI**：固有のデバイス識別子（UDI）の詳細を表示します。
- **displaylevel**：情報を表示する深さ。
- **reservation**：スマートライセンスの予約情報を表示します。
- **smart**：スマートライセンス情報を表示します。
- **status**：スマートライセンスのステータス全体を表示します。
- **summary**：スマートライセンスの概要を表示します。
- **tech-support**：スマートライセンスのデバッグ情報を表示します。
- **usage**：現在使用されているすべての権限付与のライセンス使用状況情報を表示します。



第 5 章

cnSGW-C ローリング ソフトウェア アップデート

- 機能の概要と変更履歴 （61 ページ）
- はじめに （62 ページ）
- cnSGW-C の更新 （63 ページ）
- ローリングアップグレード最適化 （75 ページ）

機能の概要と変更履歴

要約データ

表 *5:* 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	N/A
関連資料	該当なし

マニュアルの変更履歴

表 *6:* マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

はじめに

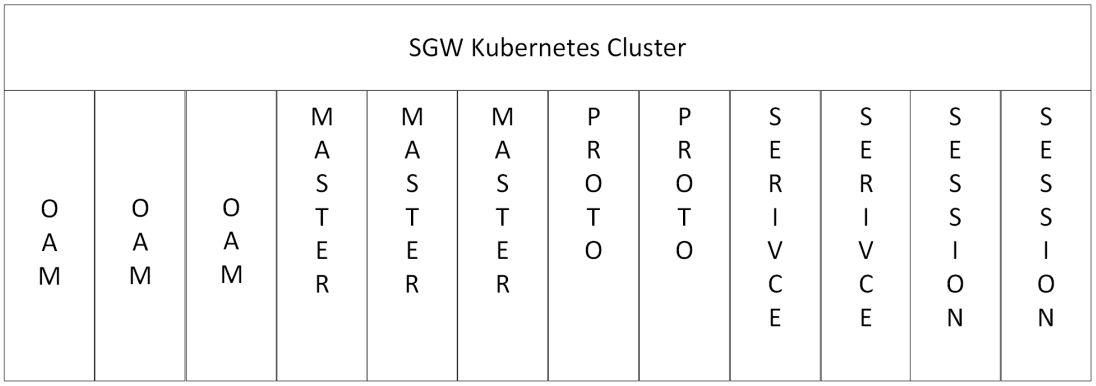
cnSGW-C は、プロトコル、サービス、セッションで構成される 3 層アーキテクチャで構成されています。各階層には、特定の機能に向けた一連のマイクロサービス（ポッド）が含まれます。これらの階層内には、Kubernetes（K8s）マスターノードとワーカーノード（運用および管理（OAM）ノードを含む）で構成される Kubernetes クラスタが存在します。

高可用性と耐障害性を確保するために、各階層に最低2つの K8s ワーカーノードが設定されています。ワーカーノードごとに複数のレプリカを設定できます。Kubernetes は、StatefulSets コントローラを使用してポッドをオーケストレーションします。ポッドには、耐障害性のために少なくとも 2 つのレプリカが必要です。

次の図は、12 のノードを持つ cnSGW-C K8s クラスタを示しています。

- 3 つのマスターノード
- 3 つの OAM ワーカーノード
- 2 つのプロトコルワーカーノード
- 2 つのサービスワーカーノード
- 2 つのセッション（データストア）ワーカーノード

図 *11 : cnSGW-C Kubernetes* クラスタ



cnSGW-C Kubernetes クラスタは、次のノードで構成されます。

- OAM ワーカーノードは、構成管理用の Ops センターポッドと、統計および重要業績評価指標（KPI）用のメトリックポッドをホストします。
- プロトコルワーカーノードは、サービスベースのインターフェイス（N11、N7、N10、N40）およびUDPベースのプロトコルインターフェイス（N4、S5/S8）に対応する cnSGW-C プロトコル関連ポッドをホストします。
- サービスワーカーノードは、セッション管理処理を実行する cnSGW-C アプリケーション関連ポッドをホストします。

- セッションワーカーノードは、サブスクライバのセッションデータを保存するデータベース関連ポッドをホストします。

cnSGW-C の更新

ローリング ソフトウェア アップデートは、古いバージョンから新しいバージョンにビルドを更新または移行するプロセス、またはアプリケーションポッドの所定の展開セットに対するパッチを更新するプロセスです。

ローリングアップデートは、ポッドインスタンスを新しいインスタンスに段階的に更新することにより、ゼロダウンタイムで実行されます。

（注） 新しいバージョンが新しいビルドバージョンまたはパッチを使用して展開されることが予想される場合、アプリケーションは使用可能である必要があります。

アップデートの範囲

同じメジャーリリース内の古いバージョンから新しいバージョンへのローリングアップデート機能がサポートされます。

- **前提条件**：更新の際は、バージョン間で以下が変更されていないことを前提としています。
  - 新旧のバージョンでサポートされている機能。
  - 既存の CLI 動作の構成の追加、削除、または変更。
  - ピア内またはポッド全体でのインターフェイスの変更。
- **Recommendations:**
  - アップデートプロセス中に構成変更をすることは推奨されません。
  - 構成の変更はすべて、アップデートプロセスの完了後に行う必要があります。
- **失敗の処理**：次の場合、ダウングレードプロセスに従って、システムを古い正常なビルドに手動でダウングレードする必要があります。
  - クラッシュなどのプロセス中の障害、およびポッド展開の失敗。
  - 新しいイベントや手順など、正常な更新後の障害。

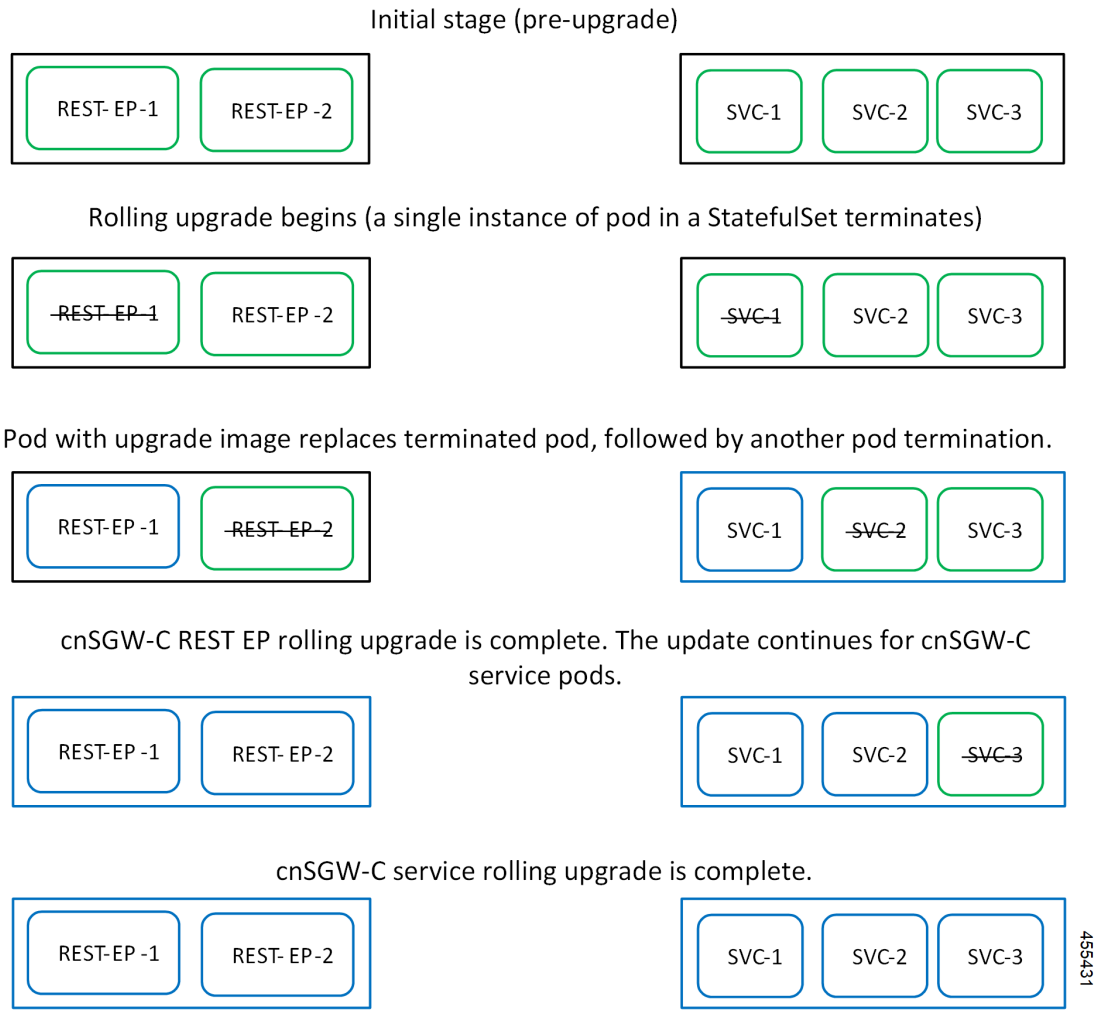
SMI Cluster Manager を使用したローリング ソフトウェア アップデート

cnSGW-C ソフトウェアアップデートまたはインサービスアップデート手順では、K8s ローリング方法を利用してポッドイメージを更新します。この方法では、進行中のプロセスが影響を受けないように、StatefulSet のポッドが順次更新されます。最初に、StatefulSet のローリングアップデートにより、単一のポッドインスタンスが終了します。終了したポッドは、更新されたイメージを持つポッドに置き換えられます。このプロセスは、StatefulSet のすべての複製がアップデートされるまで続行されます。終了するポッドは、進行中のすべてのプロセスと競合した後、正常に終了します。他のインサービスポッドはトラフィックの受信と処理を続行し、シームレスなソフトウェアアップデートを提供します。ソフトウェア アップデート プロセスは、Ops センターの CLI を使用して制御できます。

（注） 高可用性を実現するには、各ポッドに少なくとも2つのポッドが必要です。最悪のシナリオでは、ソフトウェアアップデートが進行中の間、ポッドの処理キャパシティが一時的に 50% まで低下することがあります。

次の図は、プロトコルワーカーノードでの cnSGW-C REST エンドポイントポッド（2 つのレプリカ）と、サービスワーカーノードでの cnSGW-C サービスポッド（3 つのレプリカ）の cnSGW-C ローリングアップデートを示しています。

図 *12 : cnSGW-C* ローリングアップデート



前提条件

cnSGW-C をアップグレードするための前提条件は次のとおりです。

- 稼働しているノード内のすべてのポッドを含むすべてのノード。
- cnSGW-C ソフトウェアのパッチバージョン。

（注） メジャーバージョンではローリング更新はサポートされていません。

重要 ノードの CPU 使用率が 50% 未満の場合にのみ、ローリング更新をトリガします。

- サイト内 HA サポート。

cnSGW-C ヘルスチェック

ヘルスチェックを実行して、すべてのサービスが実行され、ノードが準備完了状態になっていることを確認します。

ヘルスチェックを実行するには、次の設定を使用します。

- マスターノードにログインし、次の設定を使用します。

```
kubectl get pods -n smi
kubectl get nodes
kubectl get pod --all-namespaces -o wide
kubectl get pods -n cnsgw-wsp -o wide
kubectl get pods -n cee-wsp -o wide
kubectl get pods -n smi-vips -o wide
helm list
kubectl get pods -A | wc -l
```

重要 続行する前に、すべてのサービスが実行されており、ノードが準備完了状態になっていることを確認してください。

展開ファイルのバックアップ

バックアップ設定、ログ、および展開ファイルを作成するには、次の構成を使用します。

1. Ubuntu ユーザーとして SMI クラスタマネージャノードにログインします。
2. 展開用の新しいディレクトリを作成します。

   例：

   ```
   test@smicnsgw-cm01:~$ mkdir -p "temp_$(date +'%m%d%Y_T%H%M')" && cd "$_"
   ```

3. 新しく作成した展開ディレクトリに作業ファイルをバックアップします。
4. cnsgw 展開ファイルを解凍します。

   例：

   ```
   test@smi1cnsgw01-cm01:~/temp_08072019_T1651$ tar -xzvf cnsgw.2020.01.0-1.SPA.tgz
   ./
   ./cnsgw_REL_KEY-CCO_RELEASE.cer
   ./cisco_x509_verify_release.py
   ./cnsgw.2020.01.0-1.tar
   ./cnsgw.2020.01.0-1.tar.signature.SPA
   ./cnsgw.2020.01.0-1.tar.SPA.README
   ```

5. ダウンロードしたイメージを確認します。

   例：

   ```
   test@smi1cnsgw01-cm01:~/temp_08072019_T1651$ cat cnsgw.2020.01.0-1.tar.SPA.README
   ```

重要　次のステップに進む前に、*SPA.README* ファイルに記載されている手順に従ってビルドを確認します。

Ops センターの設定のバックアップ

Ops センターの設定をバックアップするには、次の設定を使用します。

1. Ubuntu ユーザーとして SMI クラスタマネージャノードにログインします。
2. 次の構成を使用して、SMI Ops センターの構成を **/home/ubuntu/smiops.backup** ファイルにバックアップします。

   ```
   ssh -p <port_number> admin@$(kubectl get svc -n smi | grep
   '.*netconf.*<port_number>' | awk '{ print $4 }') "show run | nomore"
    > smiops.backup_$(date +'%m%d%Y_T%H%M')
   ```

3. 次の構成を使用して、CEE Ops センターの構成を **/home/ubuntu/ceeops.backup** ファイルにバックアップします。

   ```
   ssh admin@<cee-vip> "show run | nomore" > ceeops.backup_$(date
   +'%m%d%Y_T%H%M')
   ```

4. 次の構成を使用して、cnSGW-C Ops センターの構成を **/home/ubuntu/cnSGWops.backup** ファイルにバックアップします。

   ```
   ssh admin@<cnSGW-vip> "show run | nomore" > cnSGWops.backup_$(date
   +'%m%d%Y_T%H%M')
   ```

CEE および cnSGW-C Ops Center の設定のバックアップ

マスターノードから CEE および Ops Center の設定をバックアップするには、次の設定を使用します。

1. マスターノードに Ubuntu ユーザーとしてログインします。
2. 次の設定を使用して、構成ファイルをバックアップするためのディレクトリを作成します。

   ```
   mkdir backups_$(date +'%m%d%Y_T%H%M') && cd "$_"
   ```

3. cnSGW-C Ops Center の設定をバックアップし、次の設定を使用してバックアップファイルの行数を確認します。

   ```
   ssh -p <port_number> admin@$(kubectl get svc -n $(kubectl get
   namespaces | grep -oP 'cnSGW-(\d+|\w+)') | grep <port_number> | awk
    '{ print $3 }') "show run | nomore" > cnSGWops.backup_$(date
   +'%m%d%Y_T%H%M') && wc -l cnSGWops.backup_$(date +'%m%d%Y_T%H%M')
   ```

例：

```
ubuntu@pocnSGW-mas01:~/backups_09182019_T2141$ ssh -p 2024 admin@$(kubectl get svc
-n $(kubectl get namespaces | grep -oP 'cnSGW-(\d+|\w+)') | grep <port_number> | awk
 '{ print $3 }') "show run | nomore" > cnSGWops.backup_$(date +'%m%d%Y_T%H%M') && wc
 -l cnSGWops.backup_$(date +'%m%d%Y_T%H%M')
admin@<ipv4address>'s password: cnSGW-OPS-PASSWORD
334 cnSGWops.backup
```

4. CEE Ops Center の設定をバックアップし、次の設定を使用してバックアップファイルの行数を確認します。

   ssh -p <port_number> admin@$(kubectl get svc -n $(kubectl get namespaces | grep -oP 'cee-(\d+|\w+)') | grep <port_number> | awk '{ print $3 }') "show run | nomore" > ceeops.backup_$(date +'%m%d%Y_T%H%M') && wc -l ceeops.backup_$(date +'%m%d%Y_T%H%M')

   例：

   ```
   ubuntu@pocnSGW-mas01:~/backups_09182019_T2141$ ssh -p <port_number> admin@$(kubectl
    get svc -n $(kubectl get namespaces | grep -oP 'cee-(\d+|\w+)') | grep <port_number>
    | awk '{ print $3 }') "show run | nomore" > ceeops.backup_$(date +'%m%d%Y_T%H%M')
   && wc -l ceeops.backup_$(date +'%m%d%Y_T%H%M')
   admin@<ipv4address>'s password: CEE-OPS-PASSWORD
   233 ceeops.backup
   ```

5. 次の設定を使用して、SMI Cluster Manager からの SMI Ops Center のバックアップファイルをバックアップディレクトリに移動します。

   scp $(grep cm01 /etc/hosts | awk '{ print $1 }'):/home/ubuntu/smiops.backup_$(date +'%m%d%Y_T%H%M') .

   例：

   ```
   ubuntu@pocnSGW-mas01:~/backups_09182019_T2141$ scp $(grep cm01 /etc/hosts | awk '{
   print $1 }'):/home/ubuntu/smiops.backup_$(date +'%m%d%Y_T%H%M') .
   ubuntu@<ipv4address>'s password: SMI-CM-PASSWORD
   smiops.backup                                                 100% 9346
   22.3MB/s   00:00
   ```

6. バックアップファイルの行数を確認します。

   例：

   ```
   ubuntu@pocnSGW-mas01:~/backups_09182019_T2141$ wc -l *
     233 ceeops.backup
     334 cnSGWops.backup
     361 smiops.backup
     928 total
   ```

新しい cnSGW-C イメージのステージング

更新を開始する前に新しい cnSGW-C イメージをステージングするには、次の設定を使用します。

1. 新しい cnSGW-C イメージをダウンロードして確認します。
2. Ubuntu ユーザーとして SMI クラスタマネージャノードにログインします。
3. イメージを **uploads** ディレクトリにコピーします。

```
sudo mv <cnSGW_new_image.tar> /data/software/uploads
```

（注） SMI は、**uploads** ディレクトリに存在する新しいイメージを使用して更新します。

4. **uploads** ディレクトリから処理するために、イメージが SMI によって取得されているかどうかを確認します。

   ```
   sleep 30; ls /data/software/uploads
   ```

   例：

   ```
   ubuntu@pocnSGW-cm01:~/temp_08072019_T1651$ sleep 30; ls /data/software/uploads
   ubuntu@pocnSGW-cm01:~/temp_08072019_T1651$
   ```

5. イメージが正常に取得され、処理されたかどうかを確認します。

   例：

   ```
   auser@unknown:$ sudo du -sh /data/software/packages/*
   1.6G /data/software/packages/cee.2019.07
   5.3G /data/software/packages/cnSGW.2019.08-04
   16K /data/software/packages/sample
   ```

（注） SMI は、ステージングを完了するためにイメージを **packages** ディレクトリに正常に解凍する必要があります。

ローリング ソフトウェア アップグレードのトリガー

cnSGW-C は SMI クラスタマネージャを使用して、ローリングソフトウェア更新を実行します。

SMI Cluster Manager を使用して cnSGW-C を更新するには、次の構成を使用します。

重要 開始する前に、cnSGW-C が稼働しており、現在のバージョンのソフトウェアを実行していることを確認します。

1. SMI Cluster Manager Ops Center にログインします。
2. URL から最新の tarall をダウンロードします。

   software-packages download *url*

   注：

   - **software-packages download** *url*：HTTP/HTTPS を介してダウンロードするソフトウェアパッケージを指定します。

   例：

   ```
   SMI Cluster Manager# software-packages download <url>
   ```

3. tarall がロードされているかどうかを確認します。

   software-packages list

   注：

   - **software-packages list** ：使用可能なソフトウェアパッケージのリストを指定します。

   例：

   ```
   SMI Cluster Manager# software-packages list
   [ cnSGW-2019-08-21 ]
   [ sample ]
   ```

4. 製品チャートの最新バージョンで製品リポジトリ URL を更新します。

   （注） リポジトリのURLに複数のバージョンが含まれている場合、Opsセンターは自動的に最新バージョンを選択します。

   configure
   cluster *cluster_name*
   ops-centers *app_name cnSGW_instance_name*
   repository *url*
   exit
   exit

   例：

   ```
   SMI Cluster Manager# config
   SMI Cluster Manager(config)# clusters test2
   SMI Cluster Manager(config-clusters-test2)# ops-centers cnSGW data
   SMI Cluster Manager(config-ops-centers-cnSGW/data)# repository <url>
   SMI Cluster Manager(config-ops-centers-cnSGW/data)# exit
   SMI Cluster Manager(config-clusters-test2)# exit
   ```

5. 製品チャートの最新バージョンに更新するには、次のコマンドを使用して **cluster sync** コマンドを実行します。

   clusters *cluster_name* **actions sync run**

   例：

   ```
   SMI Cluster Manager# clusters test2 actions sync run
   ```

   注：

   - **cluster** ：K8s クラスタを指定します。
   - *cluster_name* ：クラスタの名前を指定します。
   - **ops-centers** *app_name instance_name* ：製品 Ops センターとインスタンスを指定します。*app_name* は、アプリケーション名です。*instance_name* は、インスタンスの名前です。
   - **repository** *url*：チャートをダウンロードするためのローカルレジストリの URL を指定します。

- **actions** ：クラスタで実行されるアクションを指定します。
- **sync run** ：クラスタの同期をトリガーします。

重要

- クラスタ同期により、cnSGW-C Opsセンターが更新され、アプリケーションポッドが1つずつ自動的に更新されます（**helm sync** コマンドを介して）。
- 特定のポッドでローリングアップグレードをトリガーすると、cnSGW-C はそのポッドへの新しいコールのルーティングを回避します。
- cnSGW-C は、ローリングアップグレードが開始されたポッドを再起動する前に、30 秒間待機して進行中のコールを受け入れます。また、cnSGW-C は、アップグレード期間中（最大コール設定時間は 10 秒）に、進行中のすべてのコールを 30 秒以内に完全に確立します。

更新手順のモニタリング

SMI Cluster Manager Ops Center を介してステータスの更新をモニターするには、次の設定を使用します。

```
config
   clusters cluster_name actions sync run debug true
   clusters cluster_name actions sync logs
   monitor sync-logs cluster_name
   clusters cluster_name actions sync status
   exit
```

例：

```
SMI Cluster Manager# clusters test1 actions sync run
SMI Cluster Manager# clusters test1 actions sync run debug true
SMI Cluster Manager# clusters test1 actions sync logs
SMI Cluster Manager# monitor sync-logs test1
SMI Cluster Manager# clusters test1 actions sync status
```

注：

- **clusters** *cluster_name*：展開するノードに関する情報を指定します。*cluster_name* はクラスタの名前です。
- **actions**：クラスタで実行されるアクションを指定します。
- **sync run**：クラスタの同期をトリガーします。
- **sync logs**：現在のクラスタ同期ログを表示します。
- **sync status**：クラスタ同期の現在のステータスを表示します。
- **debug true**：デバッグモードを開始します。
- **monitor sync logs**：クラスタ同期プロセスをモニターします。

重要　アップグレード後は、CEE Ops Center でポッドの詳細を確認できます。ポッドの詳細については、ポッド詳細の表示 （72 ページ）を参照してください。

ポッド詳細の表示

CEE Ops センターを介して現在のポッドの詳細を表示するには、CEE Ops センター CLI で次のコマンドを使用します。

cluster pods *instance_name pod_name* **detail**

注：

- **cluster pods**：クラスタ内の現在のポッドを指定します。
- *instance_name*：インスタンスの名前を指定します。
- *pod_name*：ポッドの名前を指定します。
- **detail**：指定されたポッドの詳細を表示します。

次に、*cnSGW-data* インスタンスの *alertmanager-0* という名前のポッドの詳細を表示する例を示します。

例：

```
cee# cluster pods cnSGW-data alertmanager-0 detail
details apiVersion: "v1"
kind: "Pod"
metadata:
  annotations:
    alermanager.io/scrape: "true"
    cni.projectcalico.org/podIP: "<ipv4address/subnet>"
    config-hash: "5532425ef5fd02add051cb759730047390b1bce51da862d13597dbb38dfbde86"
  creationTimestamp: "2020-02-26T06:09:13Z"
  generateName: "alertmanager-"
  labels:
    component: "alertmanager"
    controller-revision-hash: "alertmanager-67cdb95f8b"
    statefulset.kubernetes.io/pod-name: "alertmanager-0"
  name: "alertmanager-0"
  namespace: "cnSGW"
  ownerReferences:
  - apiVersion: "apps/v1"
    kind: "StatefulSet"
    blockOwnerDeletion: true
    controller: true
    name: "alertmanager"
    uid: "82a11da4-585e-11ea-bc06-0050569ca70e"
  resourceVersion: "1654031"
  selfLink: "/api/v1/namespaces/cnSGW/pods/alertmanager-0"
  uid: "82aee5d0-585e-11ea-bc06-0050569ca70e"
spec:
  containers:
  - args:
    - "/alertmanager/alertmanager"
    - "--config.file=/etc/alertmanager/alertmanager.yml"
    - "--storage.path=/alertmanager/data"
```

```
    - "--cluster.advertise-address=$(POD_IP):6783"
    env:
    - name: "POD_IP"
      valueFrom:
        fieldRef:
          apiVersion: "v1"
          fieldPath: "status.podIP"
    image: "<path_to_docker_image>"
    imagePullPolicy: "IfNotPresent"
    name: "alertmanager"
    ports:
    - containerPort: 9093
      name: "web"
      protocol: "TCP"
    resources: {}
    terminationMessagePath: "/dev/termination-log"
    terminationMessagePolicy: "File"
    volumeMounts:
    - mountPath: "/etc/alertmanager/"
      name: "alertmanager-config"
    - mountPath: "/alertmanager/data/"
      name: "alertmanager-store"
    - mountPath: "/var/run/secrets/kubernetes.io/serviceaccount"
      name: "default-token-kbjnx"
      readOnly: true
  dnsPolicy: "ClusterFirst"
  enableServiceLinks: true
  hostname: "alertmanager-0"
  nodeName: "for-smi-cdl-1b-worker94d84de255"
  priority: 0
  restartPolicy: "Always"
  schedulerName: "default-scheduler"
  securityContext:
    fsGroup: 0
    runAsUser: 0
  serviceAccount: "default"
  serviceAccountName: "default"
  subdomain: "alertmanager-service"
  terminationGracePeriodSeconds: 30
  tolerations:
  - effect: "NoExecute"
    key: "node-role.kubernetes.io/oam"
    operator: "Equal"
    value: "true"
  - effect: "NoExecute"
    key: "node.kubernetes.io/not-ready"
    operator: "Exists"
    tolerationSeconds: 300
  - effect: "NoExecute"
    key: "node.kubernetes.io/unreachable"
    operator: "Exists"
    tolerationSeconds: 300
  volumes:
  - configMap:
      defaultMode: 420
      name: "alertmanager"
    name: "alertmanager-config"
  - emptyDir: {}
    name: "alertmanager-store"
  - name: "default-token-kbjnx"
    secret:
      defaultMode: 420
      secretName: "default-token-kbjnx"
status:
```

```
  conditions:
  - lastTransitionTime: "2020-02-26T06:09:02Z"
    status: "True"
    type: "Initialized"
  - lastTransitionTime: "2020-02-26T06:09:06Z"
    status: "True"
    type: "Ready"
  - lastTransitionTime: "2020-02-26T06:09:06Z"
    status: "True"
    type: "ContainersReady"
  - lastTransitionTime: "2020-02-26T06:09:13Z"
    status: "True"
    type: "PodScheduled"
  containerStatuses:
  - containerID:
"docker://821ed1a272d37e3b4c4c9c1ec69b671a3c3fe6eb4b42108edf44709b9c698ccd"
    image: "<path_to_docker_image>"
    imageID: "docker-pullable://<path_to_docker_image>"
    lastState: {}
    name: "alertmanager"
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: "2020-02-26T06:09:05Z"
  hostIP: "<host_ipv4address>"
  phase: "Running"
  podIP: "<pod_ipv4address>"
  qosClass: "BestEffort"
  startTime: "2020-02-26T06:09:02Z"
cee#
```

非 SMI クラスタでのローリング ソフトウェア アップデート

Helm リポジトリを設定するには、次の設定を使用します。

- cnSGW-C Ops Center にログインし、次の設定を使用します。

```
config
 helm default-repository cn
 helm repository cn
 access-token
smf-deployer.gen:AKCp5ekcX7DcBhuAmMZYfGLaHvH3E4Syr9TQDp1gjzcSjYrqsrGbXSYs5X2XYij3d9n9VfWQe

 url <old-build/new-build>
exit
```

更新の検証

正常性チェック、現在の Helm チャート、およびサブスクライバ/ピア/セッション情報は、ローリング アップデート プロセスが成功したかどうかを理解するのに役立ちます。

更新を検証するには、次の手順を実行します。

1. 展開されたポッドはすべて、更新の前後に実行状態である必要があります。

   ```
   kubectl get pods -n cn
   ```

2. Helm チャートには、適切なビルドのチャートが反映される必要があります。

現在展開されている Helm チャートを確認するには、cnSGW-C Ops Center で次のコマンドを使用します。

```
show helm charts
show running-config helm repository
```

3. 次の設定を使用して、保持の検証のためにサブスクライバ、セッション、またはピア情報を確認します。

```
show subscriber namespace sgw count all
show peers all
```

ローリングアップグレード最適化

表 7: 機能の履歴

機能名	リリース情報	説明
プロトコルPFCPポッドのローリングアップグレード最適化	2024.04.0	プロトコル（PFCP）ポッドの場合、再送信された要求を処理するために、コンバージドコア ゲートウェイでは、サービスポッドにセッションレベルの応答メッセージキャッシュが導入されます。この最適化は、アップグレード手順中のセッションと1秒あたりのコールイベント（CEPS）の損失の削減に役立ちます。

機能名	リリース情報	説明
エンドポイントポッドのローリングアップグレード最適化	2024.03.0	コンバージドコア ゲートウェイは、次のポッドのローリングアップグレード最適化サポートを提供します。 • RADIUS エンドポイントポッド • Diameter エンドポイントポッド • UDP プロキシポッド • GTPPエンドポイントポッド この最適化は、既存の最適化に対する機能拡張です。 この機能では、コンバージドコア プロファイルで既存の CLI コマンド **supported-features [app-rx-retx-cache | app-tx-retx | rolling-upgrade-all | rolling-upgrade-enhancement-infra]** が使用されます。

機能名	リリース情報	説明
ローリングアップグレード最適化	2024.02.0	コンバージド コア ゲートウェイは、次のサポートを提供します。 • アップグレード中のサービスおよびプロトコル ポッドでの再試行メカニズム • 設定ベースのローリング アップグレードの機能拡張 この最適化は、アップグレード手順中のセッションと 1 秒あたりのコールイベント（CEPS）の損失の削減に役立ちます。設定可能なローリングアップグレードの機能拡張により、変更のスムーズなロールアウトが可能になります。 この機能により、コンバージド コア プロファイルに新しい CLI コマンド **supported-features [app-rx-retx-cache | app-tx-retx | rolling-upgrade-all | rolling-upgrade-enhancement-infra]** が導入されます。 **デフォルト設定**：[Disabled]：設定が必要

コンバージド コア ゲートウェイ（CCG）ソフトウェアバージョン 2024.02.0 以降は、最適化の強化を備えたローリングアップグレードをサポートしています。ローリングアップグレードを使用すると、セッションと CEPS への影響を最小限に抑えて、すべてのポッドのグレースフル アップグレードを実行できます。最適化の強化により、必要に応じて、アップグレードプロセスの CLI コマンドを使用してローリングアップグレード機能を有効または無効にするオプションが提供され、運用上の柔軟性がもたらされます。

この機能は、次のアプリケーションレベルの機能拡張をサポートしています。

- サービスポッドのアップグレード中のプロトコルポッドにおける再試行メカニズム。
- アップグレード中のサービスポッドおよびプロトコルポッドでの一時的なセッションまたはトランザクションの処理。

- 再起動中または非アクティブなポッドを検出するためのトポロジとIPCメカニズムの変更処理。非アクティブなポッドの場合、再試行オプションはポッドの他のインスタンスに対して試行されます。

supported-features [app-rx-retx-cache | app-tx-retx | rolling-upgrade-all | rolling-upgrade-enhancement-infra] CLI コマンドを使用して、ローリングアップグレードの拡張機能を設定できます。

重要

- 既存のセッションへの影響を防ぐために、実行時にローリングアップグレード機能を有効または無効にしないことを推奨します。
- 他のコマンドオプションはすべてデバッグ目的でのみ使用できるため、**rolling-upgrade-all** オプションのみを使用することを強く推奨します。

機能説明

Converged Core Gateway（CCG）ソフトウェアバージョン 2024.02.0 以降は、最適化を追加していくローリングアップグレードをサポートしています。ローリングアップグレードを使用すると、セッションと CEPS への影響を最小限に抑えて、すべてのポッドのグレースフルアップグレードを実行できます。

この機能は、次のアプリケーションレベルの拡張機能をサポートしています。

- サービスポッドのアップグレード中のプロトコルポッドにおける再試行メカニズム。
- アップグレード中のサービスポッドおよびプロトコルポッドでの一時的なセッションまたはトランザクションの処理。
- 再起動中または非アクティブなポッドを検出するためのトポロジとIPCメカニズムの変更処理。非アクティブなポッドの場合、再試行オプションはポッドの他のインスタンスに対して試行されます。

supported-features [app-rx-retx-cache | app-tx-retx | rolling-upgrade-all | rolling-upgrade-enhancement-infra] CLI コマンドを使用して、ローリングアップグレードの拡張機能を設定できます。

重要

- 既存のセッションへの影響を防ぐために、実行時にローリングアップグレード機能を有効または無効にしないことを推奨します。
- 他のすべてのコマンドオプションはデバッグ専用であるため、**rolling-upgrade-all** オプションのみを使用することを強くお勧めします。

ローリングアップグレード最適化の仕組み

このセクションでは、さまざまなポッドのアップグレードとローリングアップグレードの手順について説明します。

ポッドのアップグレード

サービスポッド

ピアポッドは特定のポッドのアップグレードや再起動を認識するため、そのポッドとの間のトランザクションが正常に処理されるようになります。

ここでは、サービスポッドでの着信および発信メッセージの処理について説明します。

着信メッセージ（Incoming Messages）

- アップグレード中、サービスポッドの状態は、トポロジの更新を介して他のポッドに伝達されます。セッションにアフィニティが存在しない場合、そのポッドは新しいメッセージの転送ポッドとして選択されません。
- セッションでは、サービスポッドにコンテキストがない場合、これがキャッシュタイムアウト後のユーザーに対する最初のメッセージであり、アプリケーション停止メッセージがプロトコルポッドに伝達されます。そのプロトコルポッドが、他のサービスポッドインスタンスにメッセージをリダイレクトします。
- サービスポッドにセッションのコンテキストがあり、セッションまたはユーザーに対する保留中の手順やトランザクションがない場合、アプリケーション停止メッセージがプロトコルポッドに伝達されます。そのプロトコルポッドが、他のサービスポッドインスタンスにメッセージをリダイレクトします。ただし、プロトコルポッドと通信する前に、セッション状態の強制同期がCDLに対して実行されます。さらに、セッションまたはユーザーのアフィニティエントリが削除されます。

発信メッセージ（Outgoing Messages）

- セッションまたはユーザーのコールフローや手順の完了後、サービスポッドがアップグレードや再起動の通知を受信すると、セッションまたはPDUの状態とCDLとの同期が実行されます。さらに、サービスまたはユーザーのアフィニティエントリが削除され、サービスポッドインスタンスに対する後続メッセージのトリガーが許可されません。
- 進行中の既存のコールフローについては、コールを完了させるためにベストエフォートベースでトランザクションが処理されます。

プロトコルポッド

ここでは、RESTエンドポイント、GTPCエンドポイント、およびプロトコル（PFCP）エンドポイントポッドでのメッセージの処理について説明します。

REST エンドポイントポッド

ここでは、REST エンドポイントポッドでの着信および発信メッセージの処理について説明します。

着信メッセージ（Incoming Messages）

- 新しい TCP 接続の場合、入力 K8 サービスでは、アップグレードまたは再起動中に特定の REST エンドポイントポッドは選択されず、そのような要求は他のインスタンスに転送されます。
- アップグレード通知または再起動通知を受信後、既存の接続で GOAWAY フレームが送信されます。このフレームを送信すると、新しいメッセージがピアノードの新しい接続で送信されます。

発信メッセージ（Outgoing Messages）

- サービスポッドから発信要求メッセージを受信後、アプリケーション停止通知がサービスポッドに送り返されます。その結果、サービスポッドが、メッセージをトリガーする別の REST エンドポイントインスタンスを選択できます。
- 既存の一時的なメッセージの発信応答では、ベストエフォート方式でトランザクションが完了します。

GTPC エンドポイントポッド

着信メッセージ（Incoming Messages）

GTPC エンドポイントポッドの再起動中時に、着信要求メッセージの処理をサポートするために、セッションレベルの応答メッセージキャッシュがサービスポッドに追加されました。サービスポッドは、シーケンス番号、送信元 IP アドレス、送信元ポート、および要求メッセージタイプに基づいて、応答メッセージバッファを保存します。

- GTPC エンドポイントポッドの再起動が原因で回線上の応答メッセージが失われた場合、ピアで要求メッセージが再試行されます。このメッセージは、セッションレベルの応答メッセージキャッシュを使用して応答されます。
- 応答メッセージがサービスポッドで生成されない場合でも、メッセージは再送信としてサービスで検出されて適宜処理されます。

発信メッセージ（Outgoing Messages）

GTPC エンドポイントポッドの再起動中、発信要求メッセージは次の方法で処理されます。

- サービスポッドが、BGIPC を実行する際の要求のタイムアウト間隔と再送信回数を設定します。タイムアウト間隔と送信回数は、S5、S11、および S5E インターフェイスに対して定義されている N3 T3 に基づいています。
- GTPC エンドポイントポッドの代わりに、サービスポッドが送信元ポートとシーケンス番号を設定します。

- このメカニズムは、ポッドの再起動中に応答メッセージが失われた場合の発信メッセージの再送信に役立ちます。サービスポッドがシーケンス番号とポートを生成します。この場合、ピアでもメッセージが再送信として検出されるため、回線上のメッセージに不明瞭さはありません。

ノードマネージャポッド

ノードマネージャポッドの1つのインスタンスは、アップグレード中にコールを処理するために使用できます。準備状況プローブとタイマーも、アップグレードシナリオ用に設定されます。ポッドが非アクティブの場合、サービスポッドは他のノードマネージャのポッドインスタンスを再試行します。

CDL ポッド

CDLポッドには、アップグレードプロセス中にサービスを継続するための複数のレプリカがあります。アップグレードおよび再起動プロセス中の障害を最小限に抑えるために、CDLエンドポイントに対して複数の接続ストリームを使用できます。これらのプロセスでエラーが発生した場合、再試行メカニズムは別の CDL エンドポイント接続に対しても実装されます。

ローリングアップグレード最適化サポートを利用したバージョンへのソフトウェアのアップグレード

ここでは、ローリングアップグレードを実行する方法と、ローリングアップグレードの拡張機能を有効にする方法について説明します。

重要 ローリングアップグレードの手順に関連する、次の重要なガイドラインを確認してください。

- ローリングアップグレード最適化機能は、リリース 2024.02.0 以降からアップグレードする場合にのみ使用してください。

  リリース 2024.02.0 より前のバージョンからリリース 2024.02.0 以降にアップグレードする場合は、最初にシャットダウンと起動によるアップグレードを実行します。

- アップグレード後、CLI コマンドを使用して、ローリングアップグレードの拡張機能を有効にします。この場合、その後の将来のリリースへのローリングアップグレードに、利用可能な最適化が含まれるようになります。

ローリングアップグレードに関する考慮事項

SMF の両方のラック（ラック 1 とラック 2）は快晴のシナリオに設定されており、リリース 2024.02.0 より前のバージョンです。

ローリング方法を実行するには、次の手順に従います。

1. ラック 1 を雨天のシナリオに移行させ、ラック 2 を両方の GR インスタンスでアクティブのままにします。
2. ラック 1 の GR インスタンスを Standby_ERROR に移行します。

3. ラック 1 をシャットダウンします。
4. ラック 1 をリリース 2024.02 に移行するため、sync-phase ops-center を介してクラスタ同期を実行します。
5. ローリングアップグレードの拡張機能を有効にするには、推奨構成を適用します。

   ローリングアップグレードの推奨構成は次のとおりです。

   ```
   config
     profile converged-core converged_core_profile_name
     supported-features [ rolling-upgrade-all ]
     end
   ```
6. ラック 1 を開始します。
7. CDL の調整が完了するまで 30 分間待機します。
8. GR インスタンスをプライマリに切り替えて、ラック 1 をアクティブにします。
9. ラック 2 をシャットダウンします。
10. 続けて、ラック 2 でステップ 4 ～ 6 を実行します。
11. ラック 1 とラック 2 で快晴シナリオの構成を行います。

制限事項

この機能には、次の制限事項があります。

- ローリングアップグレードの間、サービスポッドは一度に 1 つずつ再起動します。このアップグレードにより、セッションが均等に再配布されます。最初に再起動するサービスポッドのセッション数が多くなります。同様に、最後に再起動したサービスポッドのセッション数が最小になります。このようなセッションの再配布により、一部のサービスポッドのメモリ要件が一時的に急増する可能性があります。セッションがサービスポッドのローカルキャッシュから削除された後、システムは期待どおりに動作します。
- サービスポッドでの一連の手順は、ローリングアップグレード中も継続されます。ただし、これらを正常に完了させるためのベストエフォートメカニズムが実装されています。

ローリングアップグレードでサポートされる機能の構成

ローリングアップグレードでサポートされている機能を有効にするには、次の構成例を使用します。

```
config
  profile converged-core cc_profile_name
     supported-features [ app-rx-retx-cache | app-tx-retx |
rolling-upgrade-all | rolling-upgrade-enhancement-infra ]
     end
```

注：

- **profile converged-core** *cc_profile_name*：コンバージド コア プロファイルの名前を指定します。このキーワードを使用すると、コンバージド コア プロファイル構成モードを開始できます。
- **supported-features [app-rx-retx-cache | app-tx-retx | rolling-upgrade-all | rolling-upgrade-enhancement-infra]**：ローリングアップグレードでサポートされている機能を有効にするには、次のいずれかのオプションを指定します。
  - **app-rx-retx-cache**：アプリケーションでの着信メッセージの再送信キャッシュを有効にします。
  - **app-tx-retx**：アプリケーションでのアウトバウンドメッセージの再送信を有効にします。
  - **rolling-upgrade-all**：**rolling-upgrade-enhancement-infra**、**app-rx-retx-cache**、および **app-tx-retxrolling** キーワードオプションで使用可能なすべてのローリングアップグレード機能を有効にします。デフォルトでは、ローリングアップグレード機能は無効になっています。

    rolling-upgrade-all は唯一の推奨オプションです。
  - **rolling-upgrade-enhancement-infra**：インフラレベルの機能を有効にします。

重要

- 既存のセッションへの影響を防ぐために、実行時にローリングアップグレード機能を有効または無効にしないことを推奨します。
- 他のすべてのコマンドオプションはデバッグ専用であるため、**rolling-upgrade-all** オプションのみを使用することを強くお勧めします。

ローリングアップグレード最適化の確認

show running-config profile amf-services コマンドを使用して、ローリングアップグレードでサポートされる機能を確認します。

show running-config amf-services コマンドの出力例を次に示します。

```
show running-config profile amf-services
amf-services am1
   supported-features [ rolling-upgrade-all ]
exit
```

OAM サポート

バルク統計

ローリングアップグレード最適化機能では、次の統計情報がサポートされています。

IPC 再試行統計情報：

再試行の原因に関するラベル「retry_cause」の追加により、「ipc_request_total」統計情報が更新されます。

CDL 統計情報：

CDL 操作に次の統計情報が追加されます。

- cdl_request_total
- cdl_response_total
- cdl_request_seconds_total
- cdl_request_duration_histgram_total

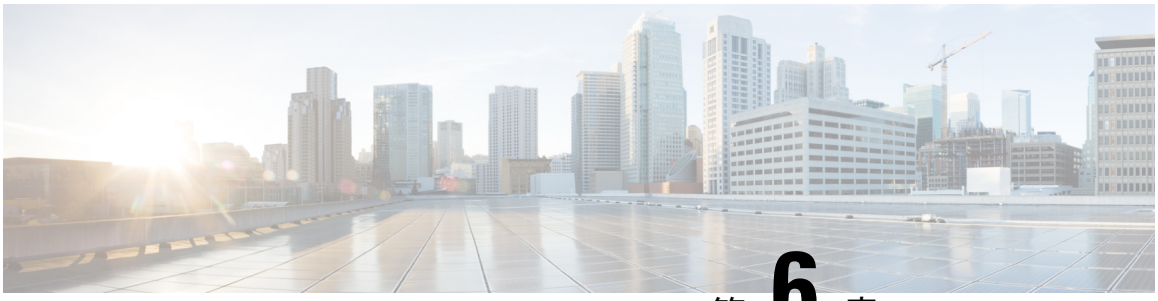
これらの CDL 統計情報は、次のフィルタで更新されます。

- **retry**：再試行メッセージを表示するために使用されます
- **method_name**：CDL 強制同期更新を表示するために使用されます

（注） 以前のリリースでは、CDL 統計情報は、フィルタ **rpc_name** を STREAM_SESSION_DB として、RPC 統計情報を介して表示されていました。リリース 2024.02 以降、CDL 統計情報は、前述の CDL 統計情報を使用する場合にのみ使用できます。

アプリケーション停止カウンタ：

App-infra でアクションを表示するために、「application_stop_action」統計情報が追加されました。これらのアクションの例には、セッションキャッシュの削除やアフィニティの削除があります。



第 6 章

ポッドとサービスの参考資料

- 機能の概要と変更履歴 （85 ページ）
- 機能説明 （86 ページ）
- ノードへのポッドの関連付け （95 ページ）
- ポッドの詳細とステータスの表示 （96 ページ）

機能の概要と変更履歴

要約データ

表 *8*: 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

改訂の詳細	リリース
初版	2020.07

機能説明

cnSGW-C は Kubernetes クラスタ戦略に基づいて構築されており、コンテナ化、高可用性、拡張性、モジュラ性、および展開の容易さというネイティブの概念を採用しています。cnSGW-C は、Kubernetes が提供するポッドやサービスなどのコンポーネントを使用します。

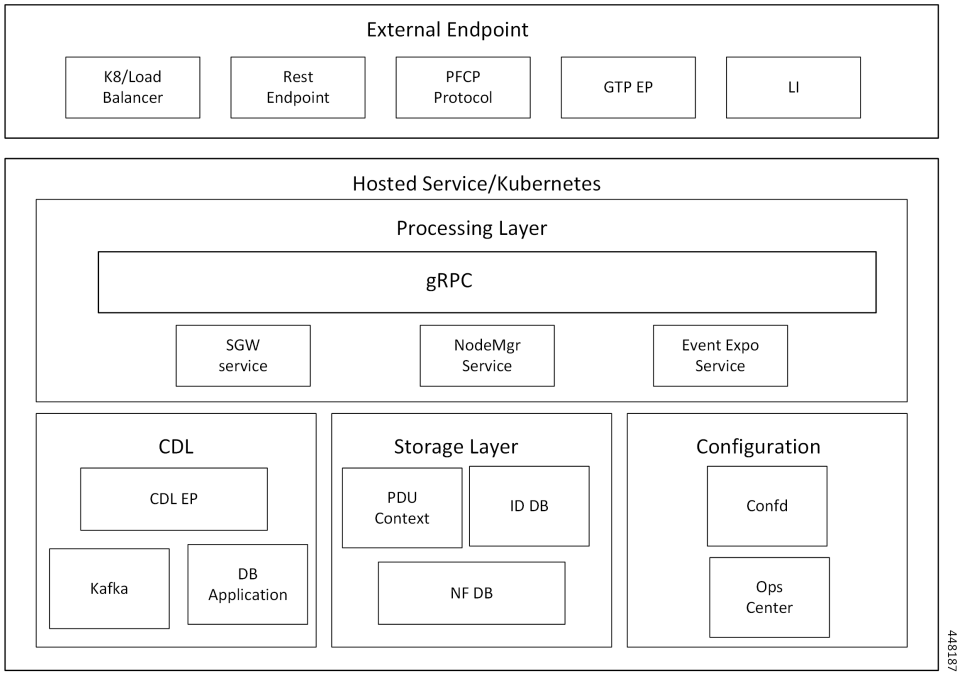
展開環境に応じて、cnSGW-Cは、設定した仮想マシン（VM）にポッドを展開します。ポッドは、ポッド内通信を担当するサービスによって動作します。ポッドをホストするマシンに障害が発生した場合、またはネットワークの中断が発生した場合、ポッドは終了または削除されます。ただし、この状況は一時的なものであり、Kubernetes は新しいポッドを作成して無効なポッドを置き換えます。

次のワークフローは、以下に関する概要レベルの情報を示しています。

- ホストマシン
- 関連するポッドとサービス
- ポッド間の相互作用

表現は、使用している展開インフラストラクチャによって異なる場合があります。

図 *13* : ポッドの通信ワークフロー



Kubernetes の展開には、クラスタ内の Kubernetes リソースを管理するための kubectl コマンドラインツールが含まれています。ポッド、ノード、およびサービスを管理できます。

Kubernetes の概念に関する一般的な情報については、Kubernetes のドキュメントを参照してください。

ポッド

ポッドは、Kubernetes クラスタで実行されるプロセスです。ポッドは、コンテナと呼ばれる細かなユニットをカプセル化します。各ポッドには、1 つ以上のコンテナを含めることができます。

Kubernetes は、物理マシンまたは仮想マシンである単一ノードに 1 つまたは複数のポッドを展開します。各ポッドには、内部 IP アドレスとポート番号を持つ個別の ID があります。ポッド内のコンテナは、ストレージリソースとネットワークリソースを共有します。

次の表に、cnSGW-C ポッドおよび共通実行環境（CEE）ポッドの名前と、割り当てたラベルに応じてそれらが展開されるホストを示します。ラベルの割り当て方法については、次の表を参照してください。

表 *9 : cnSGW-C* ポッド

ポッド名	説明	ホスト名
api-sgw-ops-center	cnSGW-C Ops センターの *confD* API ポッドとして機能します。	OAM
base-entitlement-sgw	スマートライセンス機能をサポートするために動作します。	OAM （注） 現在サポートさ
bgpspeaker	L3 ルート管理および BFD モニタリングのダイナミックルーティングをサポートするために動作します。	プロトコル
cache-pod	必要に応じて他のポッドによって使用されるシステムキャッシュ情報をサポートするために動作します。	プロトコル
cdl-ep-session-c1	CDL へのインターフェイスを提供します。	セッション
cdl-index-session-c1	セッションポッドへのキーのマッピングを保持します。	セッション
cdl-slot-session-c1	セッションデータを保存する CDL セッションポッドとして動作します。	セッション
マニュアル	ドキュメントが含まれます。	OAM
etcd-sgw-etcd-cluster	ポッドインスタンス、リーダー情報、エンドポイントなどの情報を保存する cnSGW-C OAM アプリケーションの etcd をホストします。	OAM
georeplication	キャッシュ、サイト間での ETCD レプリケーション、およびサイトロールの管理をサポートするために動作します。	プロトコル

ポッド名	説明	ホスト名
grafana-dashboard-app-infra	Grafana の app-infra メトリックのデフォルトダッシュボードが含まれます。	OAM
grafana-dashboard-cdl	Grafana の CDL メトリックのデフォルトダッシュボードが含まれます。	OAM
grafana-dashboard-sgw	Grafana の cnSGW-C サービスメトリックのデフォルトダッシュボードが含まれます。	OAM
gtpc-ep-n0	cnSGW-C の GTPC エンドポイントとして動作します。	プロトコル
kafka	CDL レプリケーションの Kafka の詳細をホストします。	プロトコル
li-ep-n0	cnSGW-C の合法的傍受エンドポイントとして動作します。	プロトコル
oam-pod	show コマンド、configuration コマンド、モニター プロトコル モニター サブスクライバなど、Ops センターのアクションを促進するためのポッドとして動作します。	OAM
ops-center-sgw-ops-center	cnSGW-C Ops センターとして機能します。	OAM
smart-agent-sgw-ops-center	cnSGW-C Ops センターのユーティリティポッドとして機能します。	OAM
nodemgr-n0	Sxa リンクの確立と管理（ハートビート）など、ノードレベルの連携動作を実行します。 UE IP アドレスや SEID などの一意の識別子を生成します。	Service
protocol-n0	アプリケーションプロトコル（PFCP、その基礎となるトランスポートプロトコルは UDP）のエンコーダおよびデコーダとして動作します。	プロトコル
rest-ep-n0	cnSGW-C は、通知クライアントとして REST-EP を使用します。	プロトコル
service-n0	cnSGW-C の主なビジネスロジックが含まれます。	Service
udp-proxy	すべての UDP メッセージのプロキシとして動作します。UDP クライアントおよびサーバーの機能を備えています。	プロトコル

ポッド名	説明	ホスト名
swift-sgw-ops-center	cnSGW-C Ops センターのユーティリティポッドとして機能します。	OAM
zookeeper	トポロジ管理のために Kafka を支援します。	OAM

CEE ポッド

詳細については、『UCC 共通実行環境 - 構成および管理ガイド』の「CEE ポッド」のトピックを参照してください。

UDP プロキシポッド

機能説明

cnSGW-C には、UP（Sxa）、MME（S11）、および PGW（S5 または S8）に対する UDP インターフェイスがあります。プロトコルレイヤポッドを利用して、これらの UDP インターフェイス上でメッセージのエンコード、デコード、および交換が行われます。

3GPP 仕様に記載されている機能を使用するには、次の手順を実行します。

- プロトコルレイヤポッドが元の送信元と宛先の IP アドレスおよびポート番号を受け取ることは必須です。ただし、着信パケットが Kubernetes（K8s）クラスタの UDP サービスに到着した後、元の IP および UDP ヘッダーは保持されません。
- 同様に、発信メッセージの場合、（ピアノードに公開される）UDP サービスの外部 IP アドレスに設定された送信元 IP が必須です。ただし、プロトコルレイヤポッドのさまざまなインスタンスが K8s クラスタのさまざまなノードから発信メッセージを送信する場合、送信元 IP は出力インターフェイスに従って選択されます。

プロトコルレイヤポッドは、前述の条件を満たすために外部 IP アドレスを設定された物理インターフェイスを持つノードで生成されます。ただし、プロトコル レイヤ ポッドを生成すると、次の結果が発生します。

- プロトコルポッドは K8s クラスタの同じノードで生成されるため、ノードレベルの HA（高可用性）を実現することはできません。そのノードに障害が発生すると、サービスが停止する可能性があります。
- プロトコルポッドには、独自の UDP クライアントおよびサーバー機能が含まれている必要があります。さらに、各プロトコルレイヤポッドには、アフィニティルールに基づいて K8s ノードにラベル付けを行う必要がある場合があります。これにより、プロトコルレイヤポッドのスケーリング要件が制限されます。

cnSGW-C では、udp-proxy と呼ばれる新しい K8s ポッドを導入することで、これらの問題に対処します。このポッドの主な目的は次のとおりです。

- udp-proxy ポッドは、あらゆる種類の UDP メッセージのプロキシとして機能します。また、UDP クライアントおよびサーバーの機能も備えています。
- プロトコルポッドは、個々のプロトコル（PFCP、GTP、Radius）のエンコーディングと復号を実行し、udp-proxy ポッドに UDP ペイロードを提供します。udp-proxy ポッドは、プロトコルポッドからペイロードを受信した後、UDP ペイロードを送信します。
- udp-proxy ポッドは、物理 IP の代わりに仮想 IP（VIP）で UDP ソケットを開きます。これにより、udp-proxy ポッドが特定の K8s ノード（VM）に対して厳密なアフィニティを持たず、UDP プロキシのノードレベルでの高可用性（HA）が実現されます。

（注） udp-proxy ポッドの 1 つのインスタンスは、デフォルトで、K8s クラスタ内のすべてのワーカーノードで生成されます。

cnSGW-C 向けの UDP プロキシ機能は、仮想 IP アドレス機能と機能的な関連性があります。

アーキテクチャ

udp-proxy ポッドは、K8s クラスタのワーカーノードに配置されます。

1. 各 K8s ワーカーノードには、udp-proxy ポッドのインスタンスが 1 つ含まれています。ただし、仮想 IP を所有するのは K8s ワーカーノードの 1 つだけです。仮想 IP を所有するワーカーノードはアクティブモードのままで、他のすべてのワーカーノードはスタンバイモードのままです。
2. アクティブな udp-proxy ポッドは、仮想 IP および指定されたポートにバインドされて、ピアノード（UPF および SGW）からの UDP メッセージをリッスンします。
3. ピアノードから受信した UDP ペイロードは、protocol、gtp-ep、または radius-ep ポッドのいずれかのインスタンスに転送されます。ペイロードは、追加処理のために同じノードまたは異なるノードのいずれかで転送されます。
4. protocol、gtp-ep、または radius-ep ポッドからの応答メッセージは、udp-proxy ポッドのアクティブなインスタンスに転送されます。udp-proxy ポッドは、対応するピアノードに応答メッセージを送り返します。
5. cnSGW-C によって開始されたメッセージは、protocol、gtp-ep、または radius-ep ポッドでエンコードされます。さらに、UDP ペイロードが udp-proxy ポッドに送信されます。最終的に、udp-proxy ポッドは完全な IP ペイロードで構成され、メッセージをピアに送信します。ピアからの応答を受信すると、UDP ペイロードは、メッセージの送信元と同じ protocol ポッドに送り返されます。

ピアが開始したメッセージのプロトコルポッドの選択

UDP プロキシポッドは、ピアノード（UPF など）によって開始されたメッセージを受信すると、プロトコルインスタンス間で負荷分散して、プロトコルポッドの任意のインスタンスを選択します。このインスタンス番号のエントリが、ピアノードの送信元 IP および送信元ポート

番号とともに保存されるため、同じ送信元 IP および送信元ポートからのメッセージが、以前に選択されたインスタンスと同じインスタンスに送信されます。

UDP プロキシの高可用性

UDP プロキシの HA モデルは、キープアライブ仮想 IP の概念に基づいています。VIP は、展開中に N4 インターフェイスに指定されます。また、キープアライブインスタンスは VIP を管理し、VIP の IP アドレスが K8s クラスタのワーカーノードのいずれかにあるインターフェイスのセカンダリアドレスとして作成されるようにします。

このワーカーノードの udp-proxy インスタンスは VIP にバインドされ、アクティブな udp-proxy ポッドのロールを担います。他のワーカーノードのすべての udp-proxy インスタンスはスタンバイモードのままです。

サービス

cnSGW-C の構成は、一連の個別のポッドで実行される複数のマイクロサービスで構成されます。これらのマイクロサービスは、cnSGW-C の展開時に展開されます。 cnSGW-C はこれらのサービスを使用して、ポッド間の通信を可能にします。別のポッドと対話する場合、サービスはポッドの IP アドレスを識別してトランザクションを開始し、ポッドのエンドポイントとして機能します。

次の表では、cnSGW-C サービスと、それらが実行されるポッドについて説明します。

表 *10 : cnSGW-C* サービスとポッド

Service Name	ポッド名	説明
base-entitlement-sgw	base-entitlement-sgw	スマートライセ めに動作します
bgpspeaker-pod	bgpspeaker	L3 ルート管理お ダイナミックル ために動作しま
datastore-ep-session	cdl-ep-session-c1	CDL セッション
datastore-notification-ep	smf-rest-ep	*smf-rest-ep* を介 に通知を送信し （注） cnSGW-C は、 REST-EP ポッド
datastore-tls-ep-session	cdl-ep-session-c1	セキュアな CD
マニュアル	マニュアル	cnSGW-C ドキ

Service Name	ポッド名	説明
etcd	etcd-sgw-etcd-cluster-0、 etcd-sgw-etcd-cluster-1、 etcd-sgw-etcd-cluster-2	名前空間内のポッ
etcd-sgw-etcd-cluster-0	etcd-sgw-etcd-cluster-0	*etcd* クラスタ間の す。
etcd-sgw-etcd-cluster-1	etcd-sgw-etcd-cluster-1	*etcd* クラスタ間の す。
etcd-sgw-etcd-cluster-2	etcd-sgw-etcd-cluster-2	*etcd* クラスタ間の す。
grafana-dashboard-app-infra	grafana-dashboard-app-infra	Grafana の app-infra トダッシュボード
grafana-dashboard-cdl	grafana-dashboard-cdl	Grafana の CDL メ ダッシュボードを
grafana-dashboard-sgw	grafana-dashboard-sgw	Grafana の cnSGW- のデフォルトダッ す。
gtpc-ep	gtpc-ep-n0	GTP-C ポッドとの ます。
helm-api-sgw-ops-center	api-sgw-ops-center	Ops センター API
kafka	kafka	Kafka メッセージを
li-ep	li-ep-n0	合法的傍受の相互
local-ldap-proxy-sgw-ops-center	ops-center-sgw-ops-center	Grafana などの他の る Ops センターの 担当します。
oam-pod	oam-pod	Ops センターでの 担当します。
ops-center-sgw-ops-center	ops-center-sgw-ops-center	cnSGW-C Ops セン
ops-center-sgw-ops-center-expose-cli	ops-center-sgw-ops-center	外部 IP アドレスで にアクセスします
smart-agent-sgw-ops-center	smart-agent-sgw-ops-center	cnSGW-C Ops セン
smf-nodemgr	smf-nodemgr	*smf-nodemgr* ポッ 当します。
smf-protocol	smf-protocol	smf-protocol ポッ 当します。

Service Name	ポッド名	説明
sgw-service	sgw-service	cnSGW-C サー 信を担当しま
swift-sgw-ops-center	swift	cnSGW-C Ops ポッドとして機
zookeeper	zookeeper	トポロジ管理の す。
zookeeper-service	zookeeper	トポロジ管理の す。

オープンポートとサービス

cnSGW-C は、通信に異なるポートを使用します。次の表は、デフォルトのオープンポートおよび関連するサービスを示しています。

表 *11* : オープンポートとサービス

Port	タイプ	Service	使用方法
22	tcp	SSH	SMI は、TCP ポートを使用して仮想マシンと通信します。
53	tcp	ドメイン	DNS ポート。
80	tcp	HTTP	SMI は、CLI、ドキュメント、および TAC への Web アクセスを提供するために TCP ポートを使用します。
111	tcp	rpcbind	オープン ネットワーク コンピューティング リモート プロシージャ コール。
179	tcp	bgp	Border Gateway Protocol（BGP）
443	tcp	SSL/HTTP	SMI は、CLI、ドキュメント、および TAC への Web アクセスを提供するために TCP ポートを使用します。
2379	tcp	etcd-client	CoreOS etcd クライアント通信。
6443	tcp	http	SMI は、ポートを使用して Kubernetes API サーバーと通信します。
7472	tcp	不明	Grafana で使用されるスピーカーです。
8083	tcp	us-srv	Kafka は REST インターフェイスに接続します。
8850	tcp	不明	udp-proxy
8879	tcp	不明	udp-proxy

Port	タイプ	Service	使用方法
9100	tcp	jetdirect	SMI は TCP ポートを使用してノードエクスポータと通信します。 ノードエクスポータは、プラガブルメトリックコレクタを備えたハードウェアおよび OS メトリック用の Prometheus エクスポータです。 これにより、メモリ、ディスク、CPU 使用率などのさまざまなマシンリソースを測定できます。
10250	tcp	SSL/HTTP	SMI は TCP ポートを使用して Kubelet と通信します。 Kubelet は Kubernetes の最下位コンポーネントです。個々のマシンで実行されている内容に関与します。 アクティブコンテナに注目するプロセスウォッチャまたはスーパーバイザです。指定されたコンテナが稼働していることを確認します。
10251	tcp	-	SMI は TCP ポートを使用して Kube スケジューラと対話します。 Kube スケジューラは Kubernetes のデフォルトのスケジューラであり、コントロールプレーンの一部として実行されます。スケジューラは、ノードが割り当てられていない新しく作成されたポッドを監視します。 スケジューラは、検出されたすべてのポッドに対して、そのポッドを実行するのに最適なノードを見つける責任を負います。
10252	tcp	apollo-relay	SMI は、この TCP ポートを使用して Kube コントローラと対話します。 Kubernetes コントローラマネージャは、Kubernetes に付属のコア制御ループを組み込むデーモンです。コントローラは、API サーバーを介してクラスタの共有状態を監視し、現在の状態を目的の状態に移行するための変更を行う制御ループです。
10256	-	HTTP	SMI は TCP ポートを使用して Kube プロキシと対話します。 Kube プロキシは、クラスタ内の各ノードで実行されるネットワークプロキシです。Kube プロキシは、ノードのネットワークルールを維持します。これらのネットワークルールにより、クラスタの内部または外部のネットワークセッションからポッドへのネットワーク通信が可能になります。
50051	tcp	不明	gRPC サービスリッスンポート。

Port	タイプ	Service	使用方法
53	udp	domain ISC BIND (Fake version: 9.11.3- 1ubuntu1.9- Ubuntu)	DNS ポート
111	udp	rpcbin	オープン ネットワーク コンピューティング リモート プロシージャ コール
2123	udp	gtpc	GTP 制御
8805	udp	pfcp	パケット転送制御プロトコル（PFCP）

ノードへのポッドの関連付け

ここでは、ポッドをノードに関連付ける方法について説明します。

クラスタを設定したら、ラベルを使用してポッドをノードに関連付けることができます。この関連付けにより、キーと値のペアに基づいて、ポッドを適切なノードに展開できます。

ポッドのラベルは、展開する必要があるノードを識別し、サービスを実行するために必要です。たとえば、必要なキーと値のペアを使用してプロトコル層のラベルを設定すると、そのキーと値のペアに一致するノードにポッドが展開されます。

1. ラベルを使用してポッドをノードに関連付けるには、次の設定を使用します。

```
config
 k8
   label
     cdl-layer
       key key_value
       value value
     oam-layer
       key key_value
       value value
     protocol-layer
       key key_value
       value value
     service-layer
       key key_value
       value value
       end
```

注：

- ラベルを設定しない場合、cnSGW-C ではデフォルトのキーと値のペアを含むラベルが想定されます。
  - **label { cdl-layer { key** *key_value* **| value** *value* **}**：CDL のキーと値のペアを設定します。
  - **oam-layer { key** *key_value* **| value** *value* **}**：OAM 層のキーと値のペアを設定します。
  - **protocol-layer { key** *key_value* **| value** *value* **}**：プロトコル層のキーと値のペアを設定します。
  - **service-layer { key** *key_value* **| value** *value* **}**：サービス層のキーと値のペアを設定します。

ポッドの詳細とステータスの表示

サービスに追加のポッドが必要な場合、cnSGW-C はポッドを作成して展開します。展開内で使用可能なポッドのリストは、cnSGW-C Ops Center で確認できます。

マスターノードから kubectl コマンドを実行して、Kubernetes リソースを管理できます。

ポッドの詳細

1. 包括的なポッドの詳細を表示するには、次のコマンドを使用します。

   kubectl get pods -n sgw *pod_name* **-o yaml**

このコマンドの出力には、次の情報を含む YAML 形式でポッドの詳細が提供されます。

- ポッドが展開されているホストの IP アドレス。
- ポッドで実行されているサービスとアプリケーション。
- ポッド内のコンテナの ID と名前。
- ポッドの IP アドレス。
- ポッドの現在の状態とフェーズ。
- ポッドが現在の状態になった時刻からの開始時刻。

次のコマンドを使用して、ポッドの詳細の概要を表示します。

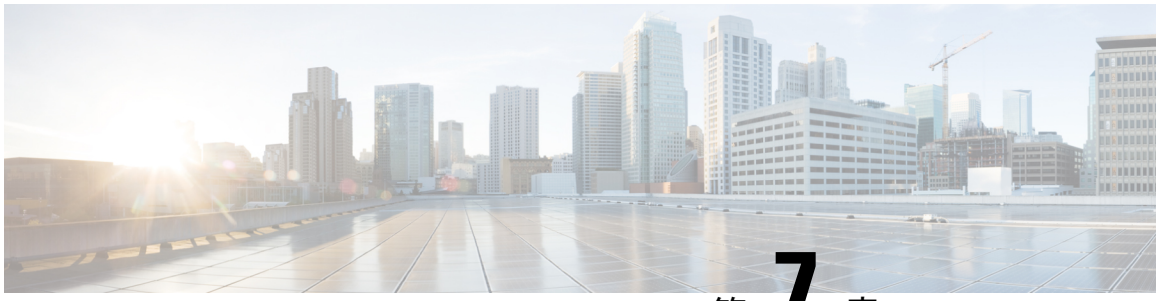
kubectl get pods -n *sgw_namespace* **-o wide**

状態

次の表で、ポッドの状態について説明します。

表 *12* : ポッドの状態

状態	説明
Running	ポッドは正常であり、ノードに展開されています。 1 つ以上のコンテナが含まれています。
Pending	アプリケーションは、ポッドのコンテナイメージを作成中です。
成功	ポッド内のすべてのコンテナが正常に終了したことを示します。これらのポッドは再起動できません。
失敗しました（Failed）	ポッド内の 1 つ以上のコンテナが終了プロセスに失敗しました。障害は、コンテナがゼロ以外のステータスで終了したか、システムによってコンテナが終了されたために発生しました。



第 7 章

3GPP RAN/NAS 原因コードのサポート

- 機能の概要と変更履歴 （99 ページ）
- 機能説明 （100 ページ）
- 機能の仕組み （101 ページ）

機能の概要と変更履歴

要約データ

表 *13* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *14* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

cnSGW-C は、*3GPP TS 29.274*、バージョン *15.4.0*、セクション *8.103*、*RAN/NAS Cause* で定義されている RAN/NAS 原因コードをサポートしています。

cnSGW-C は、MME から提供された RAN/NAS リリース原因 IE を透過的に PGW に転送し、さらに PCRF に伝達します。

（注） GTP ベースの S5/S8 および S11 がサポートされています。

次の表に、RAN/NAS 原因コードの一覧を示します。

表 *15 : RAN-NAS* 原因コード

オクテット	ビット							
	8	**7**	**6**	**5**	**4**	**3**	**2**	**1**
1	タイプ = 172（10 進数）							
2–3	長さ = n							
4	予備				インスタンス			
5	Protocol Type（プロトコル タイプ）				原因タイプ			
6 ～ m	原因値							
（m+1）～（n+4）	これらのオクテットのうち 1 以上のオクテットが存在するのは、明示的に指定されている場合のみです							

[Protocol Type] フィールドは、次のように、RAN/NAS 原因の指定された値でエンコードされます。

表 *16 : Protocol Type*（プロトコル タイプ）

Protocol Type（プロトコル タイプ）	**値（10進数）**
S1AP 原因	1
EMM 原因	2
ESM 原因	3
<spare>	4–15

[Cause Value] フィールド（および関連する RAN 原因サブカテゴリ）は、S1-AP インターフェイスを介して転送されます。フィールドは、1 オクテットのバイナリ整数としてエンコーディングされます。

表 *17*：原因タイプ

原因のタイプ	値（10 進数）
無線ネットワーク層	0
トランスポート層	1
NAS	2
プロトコル	3
その他	4
<spare>	5–15

EMM および ESM 原因については、[Cause Value] フィールドには *3GPP TS 24.301* で規定される原因値が表示されます。プロトコルが S1AP の場合、原因値には *3GPP TS 36.413* の規定値が含まれます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

このセクションでは、RAN/NAS 原因コード機能の主要なコールフローについて説明します。

ベアラー作成手順のコールフロー

このセクションでは、ベアラー作成手順のコールフローについて説明します。

図 *14 :* ベアラー作成手順のコールフロー

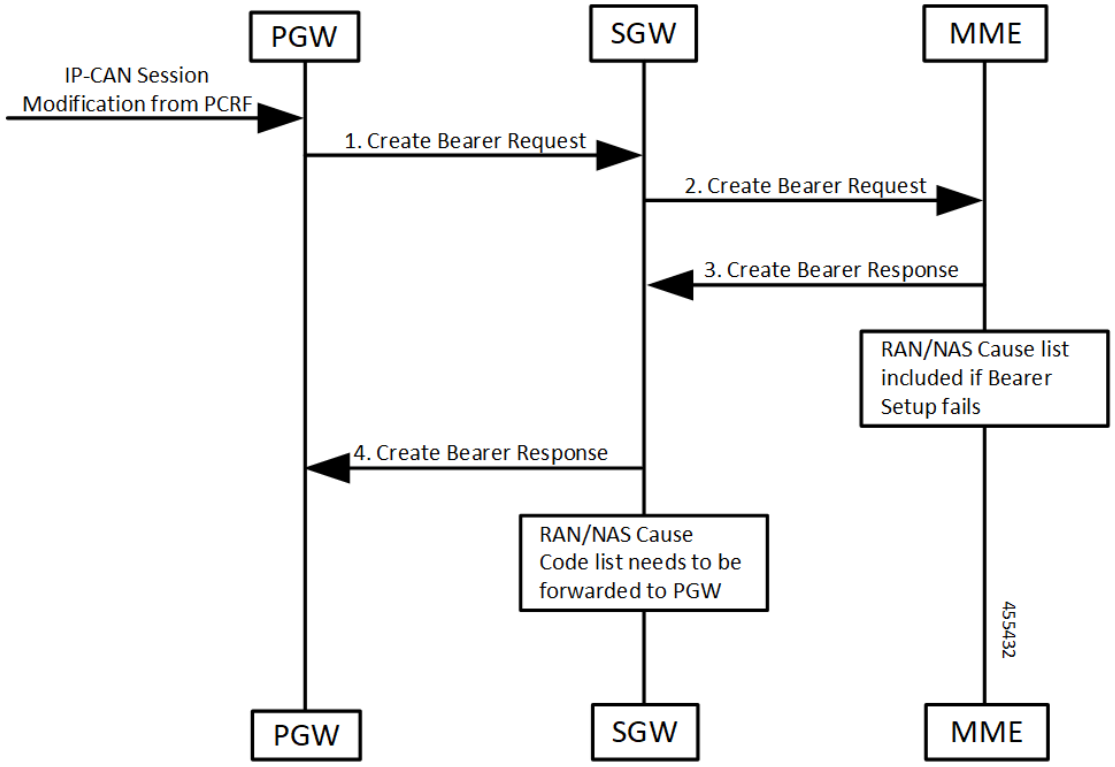


表 *18 :* ベアラー作成手順のコールフローの説明

ステップ	説明
1	PGW が PCRF から IP-CAN セッション変更要求を受信します。 PGW はベアラー作成要求メッセージを作成し、SGW（cnSGW-C）に送信します。
2	SGW（cnSGW-C）がベアラー作成要求メッセージを MME に転送します。
3	MME が SGW（cnSGW-C）に向けてベアラー作成応答メッセージを生成します。 ベアラーのセットアップが失敗した場合、RAN/原因リストが応答に含まれます。
4	SGW（cnSGW-C）は、ベアラー作成応答メッセージを PGW に転送します。 これには、RAN/NAS 原因コードのリストが含まれます。

ベアラーの更新手順のコールフロー

このセクションでは、ベアラーの更新手順のコールフローについて説明します。

図 *15* : ベアラーの更新手順のコールフロー

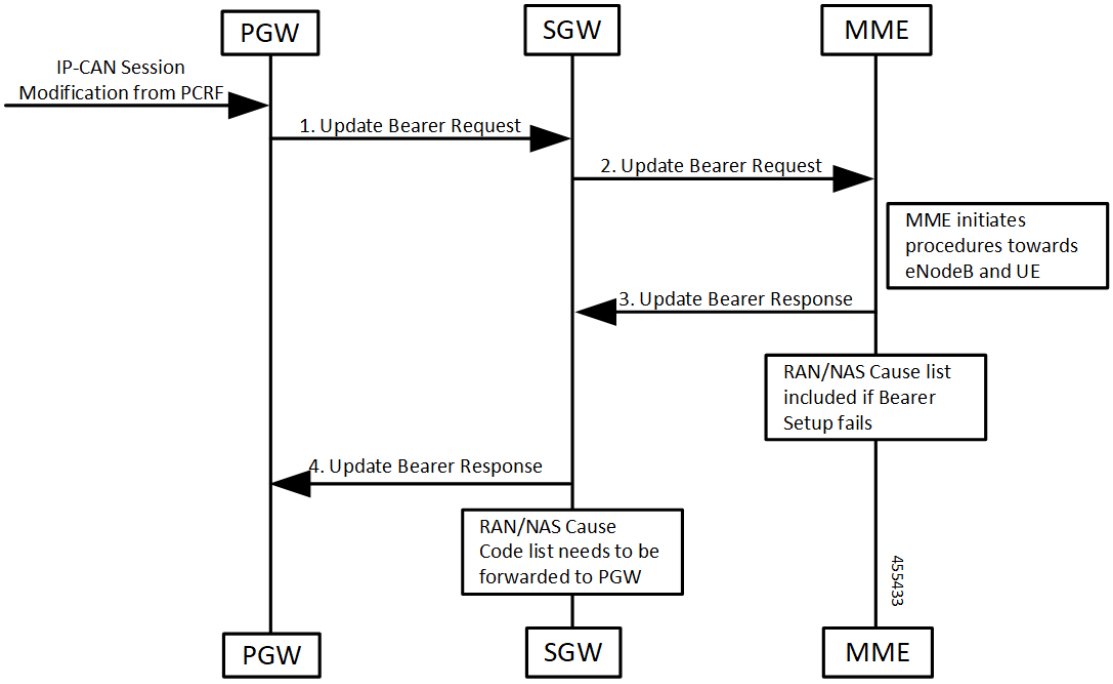


表 *19* : ベアラーの更新手順のコールフローの説明

ステップ	説明
1	PGW が PCRF から IP-CAN セッション変更要求を受信します。 PGW が SGW（cnSGW-C）へのベアラーの更新要求メッセージを作成します。
2	SGW（cnSGW-C）が MME にベアラーの更新要求メッセージを転送します。
3	MME が SGW（cnSGW-C）に向けてベアラーの更新応答メッセージを生成します。 このベアラーの変更が失敗した場合、RAN/NAS リストは応答に含まれます。
4	SGW（cnSGW-C）が PGW にベアラーの更新応答メッセージを転送します。

ベアラー削除コマンドプロシージャのコールフロー

ここでは、ベアラー削除コマンドプロシージャのコールフローについて説明します。

図 *16 :* ベアラー削除コマンドプロシージャのコールフロー

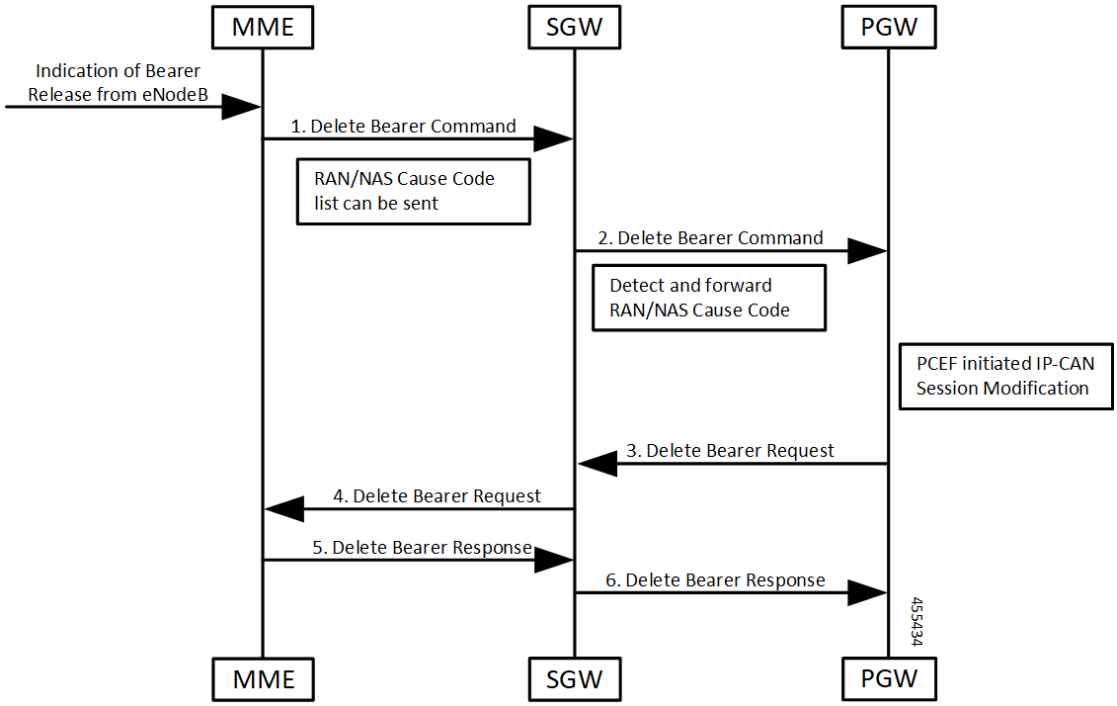


表 *20 :* ベアラー削除コマンドプロシージャのコールフローの説明

ステップ	説明
1	MME が eNodeB 要求からベアラーリリースの通知を受信します。 MME が SGW（cnSGW-C）へのベアラー削除コマンドメッセージを作成します。 これには、RAN/NAS 原因コードリストが含まれています。
2	SGW（cnSGW-C）が PGW にベアラー削除コマンドメッセージ要求を転送します。 RAN/NAS 原因コードリストを検出して転送します。
3	PGW が SGW（cnSGW-C）にベアラー削除要求メッセージを作成します。 PGW が PCEF から IP-CAN セッション変更要求を受信します。
4	SGW（cnSGW-C）が MME に向けてベアラー削除要求メッセージを生成します。
5	MME が SGW（cnSGW-C）に向けてベアラー削除応答メッセージを生成します。
6	SGW（cnSGW-C）はさらに、ベアラー削除応答メッセージを PGW に送信します。

セッション削除プロシージャのコールフロー

このセクションでは、セッションの削除手順のコールフローについて説明します。

図 *17 :* セッション削除プロシージャのコール

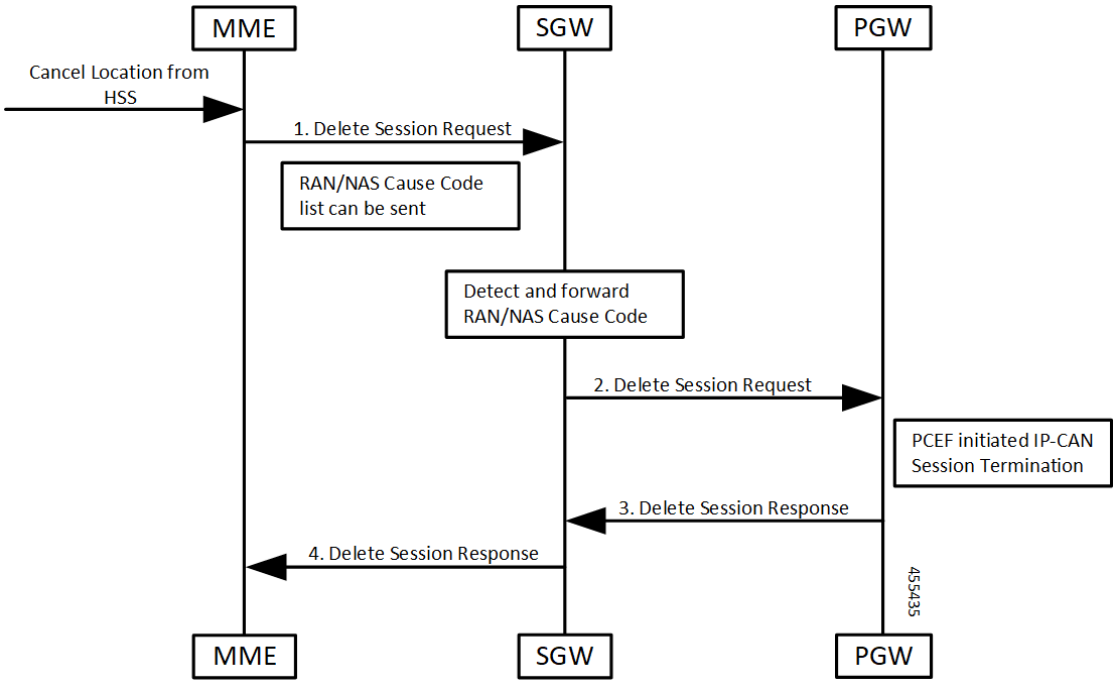
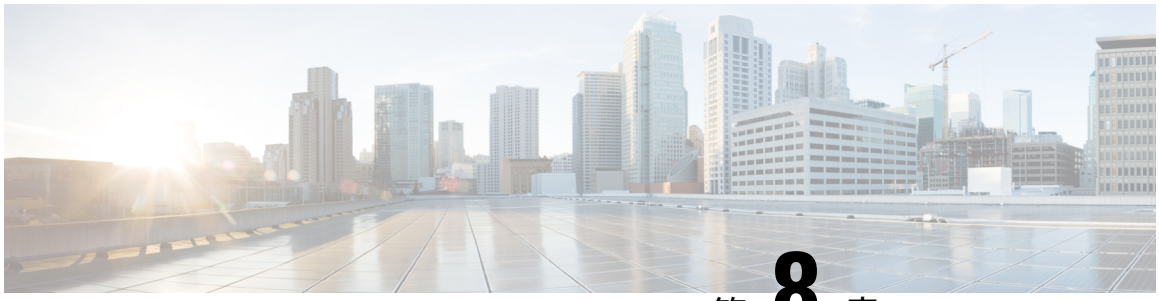


表 *21 :* セッション削除手順コールフローの説明

ステップ	説明
1	MME が HSS から場所のキャンセルの指示を受信します。 MME が SGW（cnSGW-C）へのセッション削除要求メッセージを作成します。 これには、RAN/NAS 原因コードリストが含まれています。
2	SGW（cnSGW-C）が、セッション要求メッセージの削除要求を PGW に転送します。 RAN/NAS 原因コードリストを検出して転送します。
3	PGW が、SGW（cnSGW-C）にセッション削除応答メッセージを送信します。 PGW が PCEF から IP-CAN セッション変更要求を受信します。
4	SGW（cnSGW-C）が MME に向けてセッション削除応答メッセージを生成します。



第 8 章

アクセスベアラーのリリースサポート

- 機能の概要と変更履歴 （107 ページ）
- 機能説明 （108 ページ）
- 機能の仕組み （108 ページ）

機能の概要と変更履歴

要約データ

表 *22 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *23 :* マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2020.03.0

機能説明

cnSGW-C は、アクセスベアラーのリリース（RAB）要求手順の処理をサポートしています。これはUE レベルのメッセージです。複数のPDNがあるシナリオでは、MMEはRABメッセージを 1 回だけ送信し、これがすべての PDN に適用されます。

cnSGW-C は、すべての PDN のすべてのベアラーをアイドル状態に移行させます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C は、PDN ごとに、対応するユーザープレーンに Sx Modification Request メッセージを送信します。すべてのユーザープレーン（すべての PDN）から Sx Modification Response メッセージを受信すると、cnSGW-C が応答メッセージを MME に送信します。

cnSGW-C は、CDL のすべてのベアラーの状態を IDLE として更新します。

コール フロー

このセクションでは、ベアラーリリースサポートへのアクセス機能の主なコールフローについて説明します。

アクセスベアラーのリリース（アクティブから IDLE トランザクション）コールフロー

ここでは、アクセスベアラーのリリースコールフローについて説明します。

図 *18* : アクセスベアラーのリリース（アクティブから *IDLE* トランザクション）コールフロー

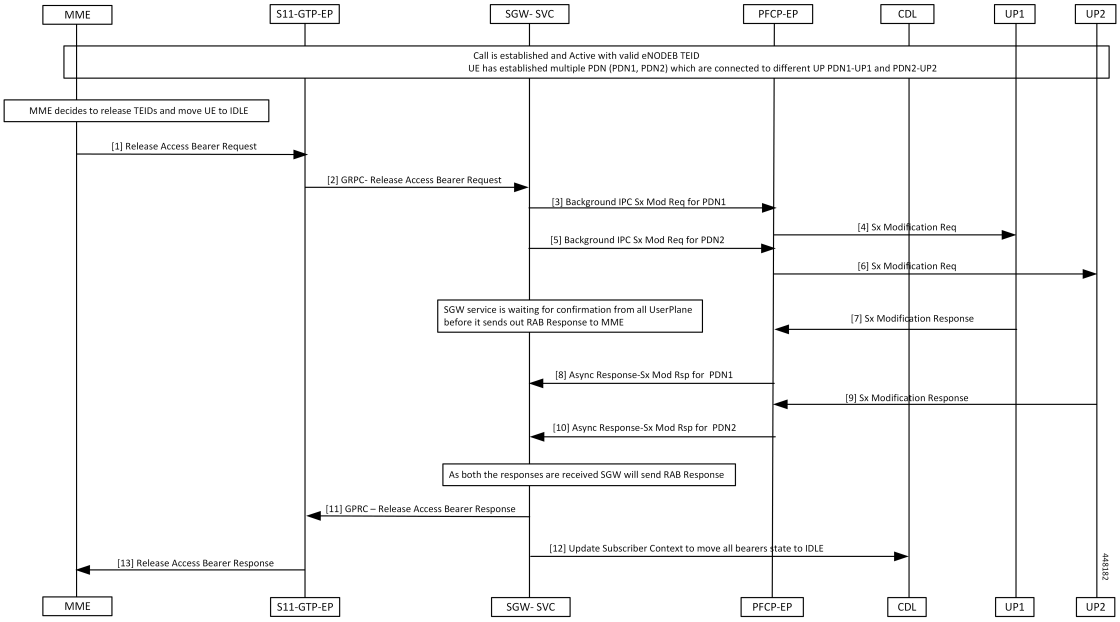
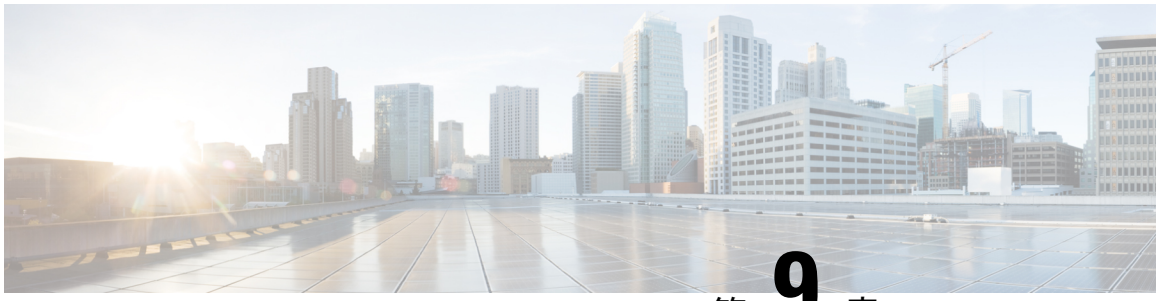


表 *24* : アクセスベアラーのリリース（アクティブから *IDLE* トランザクション）コールフローの説明

ステップ	説明
1	MME が S11-GTP-EP にアクセスベアラーのリリース（RAB）要求を送信し、UE のすべての S1-U ベアラーをリリースします。
2	S11-GTP EP が受信した UDP メッセージを復号し、gRPC に変換します。変換された gRPC メッセージは、TEID 値を使用して、この UE セッションを処理できる SGW サービスポッドに送信されます。 SGW サービスポッドは、次のアクティビティを実行します。 • ローカル入力 TEID を使用してサブスクライバ コンテキストを検索する • RAB 要求のコンテキストを検証する • UE をアイドル状態に移行する • ダウンリンク適用アクションを DROP とする Sx 変更要求メッセージを構築し、SGW-U ですべてのダウンリンクパケットをドロップする
3	SGW サービスポッドが、PDN1 のバックグラウンド IPC 非同期コールを使用して Sx Mod 要求メッセージを PFCP-EP に送信します。
4	PFCP-EP が Sx 変更要求（PDN1）メッセージを UDP プロキシを介して UPF1 に転送します。 UPF1 が Sx 変更要求（PDN1）メッセージを処理します。

ステップ	説明
5	SGW サービスポッドが、PDN2 のバックグラウンド IPC 非同期コールを使用して Sx 変更要求メッセージを送信します。 PFCP-EP が Sx 変更要求（PDN2）メッセージを UDP プロキシを介して UPF2 に転送します。
6	UPF2 が Sx 変更要求（PDN2）メッセージを処理します。
7	UPF1 が PFCP-EP に Sx 変更応答（PDN1）メッセージを送信します。
8	PFCP-EP が PDN1 の cnSGW-C サービスに非同期 Sx 変更応答メッセージを送信します。 SGW サービスポッドが PDN2 Sx 変更応答メッセージを待機します。
9	UPF2 が PFCP-EP に Sx 変更応答（PDN2）メッセージを送信します。
10	PFCP-EP が PDN2 の cnSGW-C サービスに非同期 Sx 変更応答メッセージを送信します。
11	SGW サービスポッドが、PDN（PDN1、PDN2）応答を受信した後、次を送信します。 • gRPC プロトコルを使用した S11-GTP-EP への RAB 応答メッセージ。 • CDL モジュールへのアップデート
12	SGW サービスポッドが、サブスクライバコンテキストの更新状態を CDL に送信し、すべてのベアラーが IDLE 状態に移行します。 CDL モジュールがデータベース内の情報を更新します。
13	S11-GTP-EP が MME に RAB 応答メッセージを転送します。 MME が RAB 応答メッセージを処理します。



第 9 章

APN プロファイルのサポート

- 機能の概要と変更履歴 （111 ページ）
- 機能説明 （112 ページ）
- 機能設定 （112 ページ）
- IP 送信元違反に対するアップリンクパケットの検証 （115 ページ）
- トラブルシューティング情報 （120 ページ）

機能の概要と変更履歴

要約データ

表 *25 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	[Enabled]：設定が必要
関連資料	該当なし

マニュアルの変更履歴

表 *26 :* マニュアルの変更履歴

改訂の詳細	リリース
IPv6 のサポートが導入されました。	2022.04.0
最初の導入。	2021.01.0

機能説明

この機能は、SGW（cnSGW-C）サービスのアクセスポイント名（APN）またはデータネットワーク名（DNN）プロファイルをサポートします。DNN は、Evolved Packet System（EPS）の APN に相当します。

オペレータポリシーとサブスクライバマップを使用して、cnSGW-C サービスの DNN プロファイルを決定できます。

機能設定

この機能の設定には、次の手順が含まれます。

- DNN プロファイルを設定します。詳細については、DNN プロファイルの構成 （112 ページ） を参照してください。
- ネットワーク要素プロファイルを設定します。詳細については、ネットワーク要素プロファイルの設定 （112 ページ）を参照してください。

DNN プロファイルの構成

この機能を設定するには、次の構成を使用します。

```
config
  profile dnn dnn_name
    upf-selection-policy upf_select_name
      dnn dnn_name network-function-list network_function_list
      end
```

注：

- **dnn** *dnn_name*：DNN プロファイル名を指定します。文字列で指定する必要があります。
- **upf-selection-policy** *upf_select_name*：UPF 選択ポリシー名を指定します。文字列で指定する必要があります。
- **network-function-list** *network_function_list*：選択した DNN プロファイルを送信するネットワーク機能のリストを指定します。文字列で指定する必要があります。

ネットワーク要素プロファイルの設定

ネットワーク要素プロファイルはピア IP（UPF）プロファイルを表し、次の設定があります。

- ピアアドレスとポートの設定
- ピアによってサポートされる DNN または APN。この設定は、UPF の選択に役立ちます。

UPF の選択では、優先順位とキャパシティパラメータを考慮します。

upf-group-profile は、そのプロファイルが属する UPF グループを示します。

この機能を設定するには、次の設定を使用します。

```
config
  profile network-element upf upf_name
    node-id  node_id_value
    n4-peer-address ipv6 ipv6_address
    n4-peer-address ipv4 ipv4_address
    n4-peer-port port_number
    dual-stack-transport { true | false }
    dnn-list dnn_list
    capacity capacity_value
    priority priority_value
    upf-group-profile upf_group_name
    end
```

注：

- **network-element**：ピアネットワーク要素を指定します。
- **upf** *upf_name*：UPF ピア名を指定します。
- **node-id** *node_id_value*：UPF ノードのノード ID を指定します。
- **n4-peer-address ipv4** *ipv4_address*：IPv4 アドレスを指定します。
- **n4-peer-address ipv6** *ipv6_address*：IPv6 アドレスを指定します。
- **n4-peer-port** *port_number*：N4 ピアポート番号を指定します。これは、0 ー 65535 の範囲の整数である必要があります。
- **dual-stack-transport { true | false }**：IPv6 または IPv4 アドレスを指定できるデュアルスタック機能を有効にします。この機能を有効にするには、true を指定します。
- **dnn-list** *dnn_list*：UPF ノードでサポートされている DNN リストを指定します。
- **capacity** *capacity_value*：他の UPF に対するキャパシティを指定します。これはロードバランシングに使用されます。これは、0 ー 65535 の範囲の整数である必要があります。デフォルト値は 10 です。
- **priority** *priority_value*：他の UPF に対する静的優先順位を指定します。これはロードバランシングに使用されます。これは、0 ー 65535 の範囲の整数である必要があります。デフォルト値は 1 です。
- **upf-group-profile** *upf_group_name*：UPF グループプロファイル名を指定します。文字列で指定する必要があります。

設定の変更による影響

次の表は、既存のコール、新しい PDN、または新しいサブスクライバに対する設定変更動作の影響を示します。

変更内容	cnSGW-C の既存コール	cnSGW-Cの新しいPDNまたは新しいサブスクライバ
APN プロファイルを削除	影響なし	変更内容に基づいて以下に対して新しい設定が適用されました。 • サブスクライバポリシー • オペレータポリシー • DNN ポリシー • DNN プロファイル
オペレータポリシーで APN プロファイル名を変更	影響なし	変更内容に基づいて以下に対して新しい設定が適用されました。 • サブスクライバポリシー • オペレータポリシー • DNN ポリシー • DNN プロファイル

IP 送信元違反に対するアップリンクパケットの検証

表 *27 :* 機能の履歴

機能名	リリース情報	説明
IP 送信元違反に対するアップリンクパケットの検証	2024.03.0	IP 送信元違反検出機能が有効な場合、cnSGW は、受信パケットの送信元 IP アドレスを有効なユーザー機器（UE）の IP アドレスと照合することで、不正なパケットを検証します。 cnSGW はセッション作成要求または応答から UE の IP アドレスを取得し、Sx セッション確立要求メッセージで UPF に送信するための IP 送信元違反情報要素（IE）を含めます。この IE は、無効な送信元 IP アドレスを持つパケットに対して UPF が動作するように設定されたアクション（IGNORE または DISCARD）を示しています。 この機能により、権限のない関係者へのデータの漏洩を防止することで、ネットワークセキュリティとサブスクライバのプライバシーが強化されます。この機能は、合法的傍受要件に準拠することにより、サービスプロバイダーの法的および規制的問題のリスクをさらに軽減します。 **CLI 導入**：DNN プロファイルの **ip source-violation [ignore | discard]**。 **デフォルト設定**：[Disabled]：有効にするには設定が必要。

デフォルトでは、アップリンクパケットの IP 送信元違反検出は無効になっています。無効なパケットに対して IGNORE または DISCARD アクションを設定して、この機能を有効化する必要があります。有効にすると、cnSGW はセッション作成要求または応答から UE IP アドレスを取得し、UPF に送信します。無効な送信元 IP アドレスを持つパケットに対して UPF が実行する必要があるアクションは、Sx セッション確立要求メッセージの IP 送信元違反 IE を使用して示されます。

ネットワークオペレータの特定の要件とポリシーに応じて、IP 送信元違反検出機能に対して次のオプションから選択できます。

- [IGNORE]：ネットワークに厳しいセキュリティ要件がなく、無効な送信元 IP アドレスを無視して運用の柔軟性を維持する場合はこのオプションを選択します。統計を蓄積しながら無効なパケットを無視することで、トラフィックをすぐに廃棄することなく、無効な送信元 IP アドレスの発生をモニタリングおよび分析できます。送信元 IP アドレスが無効であってもパケットを転送することにより、無効なトラフィックの影響が最小限であるシナリオにおいて、ネットワークは一貫したユーザーエクスペリエンスを維持できます。
- [DISCARD]：ネットワークに厳しいセキュリティ要件があり、無効な送信元 IP アドレスを持つパケットを廃棄して、IP スプーフィングなどの潜在的なセキュリティ脅威を防止する場合はこのオプションを選択します。無効な送信元 IP アドレスを持つパケットを破棄することで、有効なトラフィックのみが処理および転送され、ネットワークの整合性が維持されます。無効なパケットを破棄することで、ネットワークは、誤った送信元 IP アドレスを持つパケットの処理に起因する潜在的な誤ルーティングを回避できます。

（注） この機能が無効になっている場合、IP 送信元違反 IE は Sx セッション確立要求メッセージに含まれません。

無効な送信元 IP アドレスを持つパケットに対しては、パケットが無視されたか破棄されたかに関わらず、対応する統計が蓄積されます。

この機能は、**ip source-violation [ignore | discard]** CLI コマンドを使用して設定できます。

UPF でのアップリンクパケットの IP 送信元違反の有効化の詳細については、UPF 設定および管理ガイドの「アップリンクパケットの IP 送信元違反の有効化」を参照してください。

IP 送信元違反に対するアップリンクパケットの検証の仕組み

IP 送信元違反情報要素（IE）のデータ構造により、cnSGW から UPF への無効な送信元 IP アドレスを持つパケットに対して実行されるアクションが指示されます。この IE は Sx Session Establishment Request メッセージの一部であり、IP 送信元違反検出機能が有効になっている場合に含まれます。

表 *28 :* 無効な送信元 *IP* アドレスを持つアップリンクパケットの処理

無効になっている場合	有効になっている場合
IP 送信元違反 IE は、Sx Session Establishment Request メッセージに含まれません。	次のいずれかのオプションを設定します。 • IGNORE • 無効な送信元 IP アドレスを持つパケットが検証され、適用可能なポリシーに従ってネットワークを介して転送されます。 • cnSGW に保持されている対応する統計は、無効な送信元 IP アドレスを検出したときに増分されます。 • 無視されたパケットはネットワーク トラフィック フローに存在し続け、有効なパケットとして処理されます。 • DISCARD • 無効な送信元 IP アドレスを持つパケットが特定され、ドロップされます。 • cnSGW に保持されている対応する統計は、パケットを破棄するアクションに対して増分されます。 • 破棄されたパケットは、ネットワークデバイスによってそれ以上処理されず、保存も、ログ記録も、転送もされません。

IP送信元違反に対するアップリンクパケットの検証の有効化と無効化

DNN プロファイルで IP 送信元違反の検出機能を有効または無効にすることができます。CLI コマンドを使用すると、ネットワーク内で無効な送信元 IP アドレスを持つパケットを処理する方法を構成できます。

手順

ステップ 1 プロファイル DNN モードにログインし、DNN プロファイル名を入力します。

例：

```
[sgw] smf(config)# profile dnn <profile name>
```

ステップ 2 **ip source-violation** コマンドを入力して、IP 送信元違反の検出を設定します。

例：

```
[sgw] smf(config)# profile dnn <profile name> ip source-violation
```

ステップ 3 **ip source-violation** コマンドの *iignore* または *discard* オプションを選択します。

例：

```
[sgw] smf(config-dnn-intershat)# ip source-violation [ ignore | discard ]
```

DNN プロファイルで IP 送信元違反を無効にするには、**no ip source-violation** CLI コマンドを使用します。

Ignore オプションを入力すると、UPF はパケットの IP 送信元違反をチェックしません。*discard* オプションを入力すると、UPF はエラーのあるパケットを破棄します。

次のタスク

この機能が構成されているかどうかを確認するには、次のいずれかのオプションを使用します。

- DNN プロファイルでのアップリンクパケット送信元検証の確認 （118 ページ）
- NF サービスのアップリンクパケット送信元の検証の確認 （119 ページ）

DNN プロファイルでのアップリンクパケット送信元検証の確認

DNN プロファイルの手順を使用して、設定されたアクションで IP 送信元違反が無効か有効かを確認します。

手順

DNN プロファイルに **show running-config profile dnn** *dnn_name* を入力します。

例：

```
show running-config profile dnn intershat
```

次に、*discard* オプションを設定した **show running-config profile dnn** *dnn_name* CLI コマンドの出力例を示します。

```
show running-config profile dnn intershat
 profile dnn intershat
 dns primary ipv4 209.165.200.229
 dns primary ipv6 66:66:1::aa
 session type IPV4 allowed [ IPV6 IPV4V6 ]
 upf apn intershat
 ip source-violation discard
exit
```

NF サービスのアップリンクパケット送信元の検証の確認

次の手順を使用して、IMSI、MEI、電話番号、および特定のサブスクライバの IP 送信元違反機能の構成などのサブスクライバの詳細を確認します。

手順

show subscriber nf-service sgw imsi *imsi_value* を入力します。

例：

```
show subscriber nf-service sgw imsi ABX
```

次に、*ignore* オプションを構成した **show subscriber nf-service sgw imsi** *imsi_value* CLI コマンドの出力例を示します。

```
show subscriber nf-service sgw imsi ABX
subscriber-details
{
  "subResponses": [
    {
      "status": true,
      "genericInfo": {
        "imsi": "imsi-204163553638360",
        "mei": "imeisv-3590730650683317",
        "msisdn": "msisdn-31638577770",
        "accessType": "EUTRAN",
        "plmnId": {
          "mcc": "310",
          "mnc": "260"
        },
        "sgwProfileName": "cn-sgw",
        "unAuthenticatedImsi": "No"
      },
      "s11cInterfaceInfo": {
        "sgwTeid": "[0x1265a5a7] 308651431",
        "sgwIPv4Address": "10.210.81.0",
        "mmeTeid": "[0xa61f5cc] 174192076",
        "mmeIPv4Address": "172.57.38.19"
      },
      "pdnInfoList": {
        "totalPdn": 1,
        "pdnInfo": [
          {
            "pdnId": "PDN-1",
            "apn": "smartsites.t-mobile",
            "attachType": "Initial Attach",
…
…
            "plmnType": "ROAMER",
            "s5ePeerType": "ROAMER",
            "collocatedSub": "NonCollocated"
            "ipSrcViolation": "ignore"
          }
        ]
      }
    }
```

```
  ]
}
```

トラブルシューティング情報

ここでは、この機能のトラブルシューティング情報について説明します。

設定エラー

ここでは、APN プロファイルの設定中に cnSGW-C から報告される可能性のあるエラーについて説明します。

```
show config-error | tab
ERROR COMPONENT   ERROR DESCRIPTION
-------------------------------------------------------------------------------------------------
SGWProfile        Subscriber policy name : sub_policy in profile sgw1 is not configured
SubscriberPolicy  Operator policy : op_policy1 under subscriber policy sub_policy2 is
not configured
OperatorPolicy    Dnn policy name : dnn_policy1 in operator policy op_policy2 is not
configured
DnnPolicy         Dnn profile name : dnn_profile1 in dnn policy dnn_policy2 is not
configured
DnnProfile       UPF selection policy name : upf_sel_policy1 in dnn profile dnn_profile2
 is not configured
```

第 10 章

変更通知要求の処理

- 機能の概要と変更履歴 （121 ページ）
- 機能説明 （122 ページ）
- 機能の仕組み （122 ページ）
- OAM のサポート （124 ページ）

機能の概要と変更履歴

要約データ

表 *29* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *30* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

変更通知メッセージが cnSGW-C で開始され、ユーザーロケーション情報（ULI）およびユーザー CSG 情報（UCI）の更新の変更を示します。これらの更新が有効な場合、cnSGW-C CDR が開始されます。変更通知には、cnSGW-C および ISGW に固有のセカンダリ RAT 使用量 IE が含まれる場合があります。cnSGW-C は RAT 使用情報を保存し、後続の CDR メッセージで使用情報を送信します。

標準準拠

この機能は、次の標準仕様に準拠しています。

- 3GPP TS 29.274「3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3」

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C ネットワーク関数は、次のアプローチを使用して変更通知要求を処理します。

- 接続要求（CNREQ）で ULI または UCI の変更が有効であれば、関連するパケットデータネットワーク（PDN）が更新されます。
- cnSGW-C はクエリ URR を開始して最新の使用状況情報を取得し、次の場合に cnSGW-C CDR を生成します。
  - ULI が変更された。
  - 課金と ULI トリガーが有効になっている。

課金の構成の詳細については、「SGW 課金のサポート」の章を参照してください。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

変更通知要求のコールフロー

このセクションでは、変更通知の要求と応答のコールフローについて説明します。

図 *19* : 変更通知要求のコールフロー

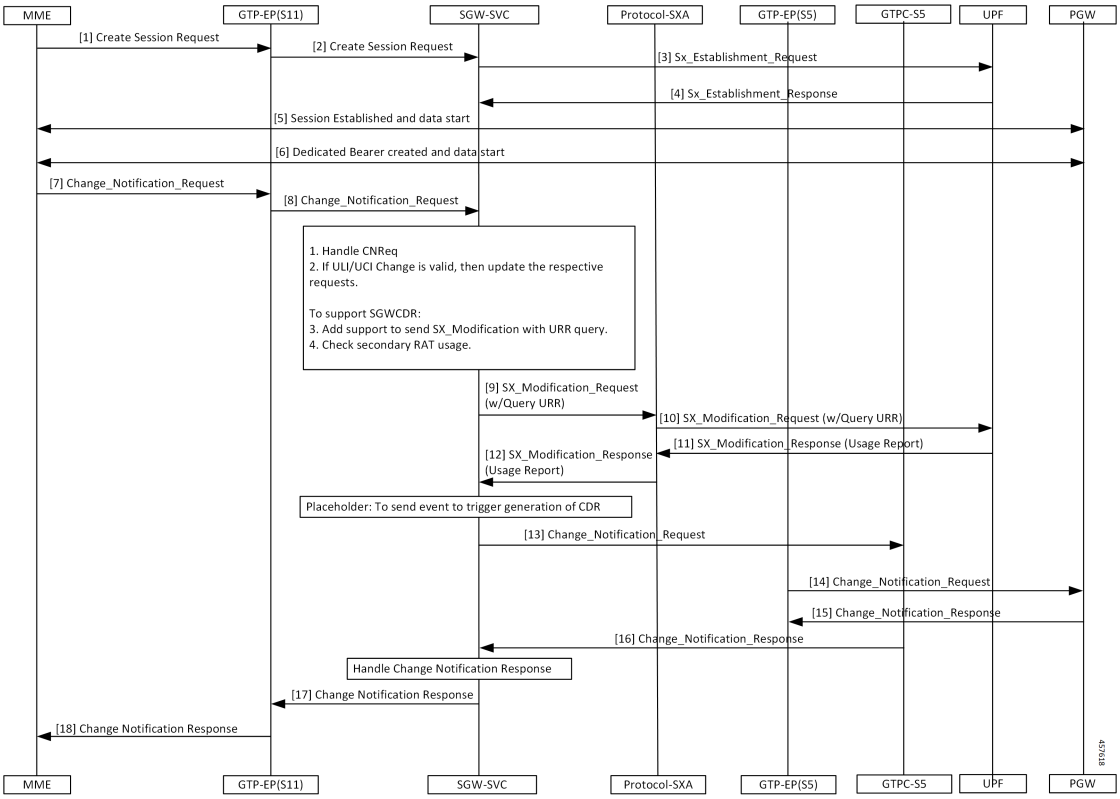


表 *31* : 変更通知要求のコールフローの説明

ステップ	説明
1	MME がセッション作成要求を GTP-EP（S11）に送信します。
2	GTP-EP（S11）がセッション作成要求を SGW-SVC に転送します。
3	SGW-SVC が Sx 確立要求を UPF に送信します。
4	UPF が、SGW-SVC に向けられた SX 確立応答で要求に応答します。
5	MME と PGW がセッションを確立し、データの交換を開始します。
6	MME と PGW が専用ベアラーを作成し、データの交換を開始します。
7	MME が変更通知要求を GTP-EP に送信します。

ステップ	説明
8	GTP-EP が変更通知要求を SGW-SVC に転送します。 接続要求（CNREQ）で ULI または UCI の変更が有効な場合、PDN が更新されます。 GTP-EP は、セカンダリ RAT の使用状況を確認した後、URR クエリを含む Sx 変更要求を送信します。
9	SGW-SVC が、URR クエリを含む Sx 変更要求を Proto-SXA に送信します。
10	Proto-SXA が、URR クエリを含む Sx 変更要求を UPF に転送します。
11	UPF はこの要求に対し、Proto-SXA に対する使用状況レポートを含む Sx 変更応答で応答します。
12	Proto-SXA が、使用状況レポートを含む Sx 変更応答を SGW-SVC に転送します。
13	SGW-SVC が変更通知要求を GTPC-S5 に送信します。
14	GTPC-S5 が変更通知要求を PGW に転送します。
15	PGW が GTPC-S5 への変更通知応答で応答します。
16	GTPC-S5 が変更通知応答を SGW-SVC に転送します。
17	SGW-SVC が GTP-EP（S11）に変更通知応答を送信します。
18	GTP-EP（S11）が変更通知応答を MME に転送します。

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計サポート

変更通知フィルタには、通知が呼び出される変更要求のステータスが表示されます。以下はサンプル統計であり、参照目的でのみ提供されます。

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="change_notification",status="attempted",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="change_notification",status="success",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="initial_attach",status="attempted",sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="initial_attach",status="success",sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="change_notification",status="attempted",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="change_notification",status="success",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="initial_attach",status="attempted",sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="initial_attach",status="success",sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",

sgw_procedure_type="modify_bearer_req_initial_attach",status="attempted",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service"
```

```
,sgw_procedure_type="modify_bearer_req_initial_attach",status="success",sub_fail_reason=""}
 1
```

第 11 章

サブスクライバ要求のクリア

- 機能の概要と変更履歴 （127 ページ）
- 機能説明 （127 ページ）
- 機能の仕組み （128 ページ）

機能の概要と変更履歴

要約データ

表 *32* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *33* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2020.04

機能説明

cnSGW-C はサブスクライバのクリアまたは Ops センターからの PDN 要求を処理します。

サブスクライバのクリア要求は、ローカル消去およびリモートシグナリングの手順を使用して、特定の IMSI またはすべての IMSI のサブスクライバの管理クリアを開始します。

OAM クエリに基づいて、cnSGW-C は REST-EP で加入者通知メッセージを受信し、SGW サービスに向けて加入者要求のクリアメッセージをトリガします。

標準準拠

この機能は、次の標準仕様に準拠しています。

- 3GPP TS 23.401「General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access」
- 3GPP TS 23.214「Architecture enhancements for control and user plane separation of EPC nodes」
- 3GPP TS 29.274「3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3」
- 3GPP TS 29.244「Interface between the Control Plane and the User Plane nodes」

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C は、管理者が開始した、消去オプションが「true」の削除要求を受信すると、ユーザープレーンへの Sx シグナリングを開始し、次のメッセージを交換します。

1. SGW がユーザープレーンに Sx セッション削除要求を送信します。
2. ユーザープレーンが Sx セッション削除応答を SGW に送信します。

cnSGW-C は、消去オプションが「false」の削除要求を受信すると、ユーザープレーンへの Sx シグナリングと MME および PGW への GTP-C シグナリングを実行します。cnSGW-C は、ユーザープレーン、MME、および PGW と次のメッセージを交換します。

1. SGW が Sx セッション変更要求をユーザープレーンに送信します。
2. ユーザープレーンが Sx セッション変更応答を SGW に送信します。
3. SGW が MME にベアラー作成要求を送信します。
4. SGW が PGW にセッション削除要求を送信します。
5. MME が SGW にベアラー削除応答を送信します。
6. PGW が SGW にセッション削除応答を送信します。
7. SGW が Sx セッション削除要求をユーザープレーンに送信します。
8. ユーザープレーンが Sx セッション削除応答を SGW に送信します。

cnSGW-C がセッション削除要求を PGW に送信し、ベアラー削除要求を MME に送信します。両方のリモートピアからの応答を受信した後、cnSGW-C はユーザープレーンに Sx セッション削除要求を送信してセッションをクリアします。

サポートされる clear コマンド

cnSGW-C は、次の clear コマンドをサポートしています。

表 *34* : サポートされる *clear* コマンド

サポートされる clear コマンドオプション	GTP-C シグナリング（MME/PGW 向け）	Sx シグナリング（UP 向け）	影響（サブスクライバ/PDN）
clear sub all **clear sub all purge false**	対応	対応	すべてのサブスクライバ
clear sub all purge true	非対応	対応	すべてのサブスクライバ
• **clear sub namespace sgw imsi** *imsi_val* • **clear sub namespace sgw imsi***imsi_val***purge false**	対応	対応	IMSI が *imsi_val*のサブスクライバ
clear sub namespace sgw imsi*imsi_val***purge true**	非対応	対応	IMSI が *imsi_val*のサブスクライバ
clear sub namespace sgw imsi*imsi_val***ebi***ebi_value*	対応	対応	IMSI が *imsi_val* で、デフォルトの ebi が *ebi_value* の PDN

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

クリア PDN コールフロー

ここでは、クリア PDN コールフローについて説明します。

図 *20 :* クリア *PDN* コールフロー

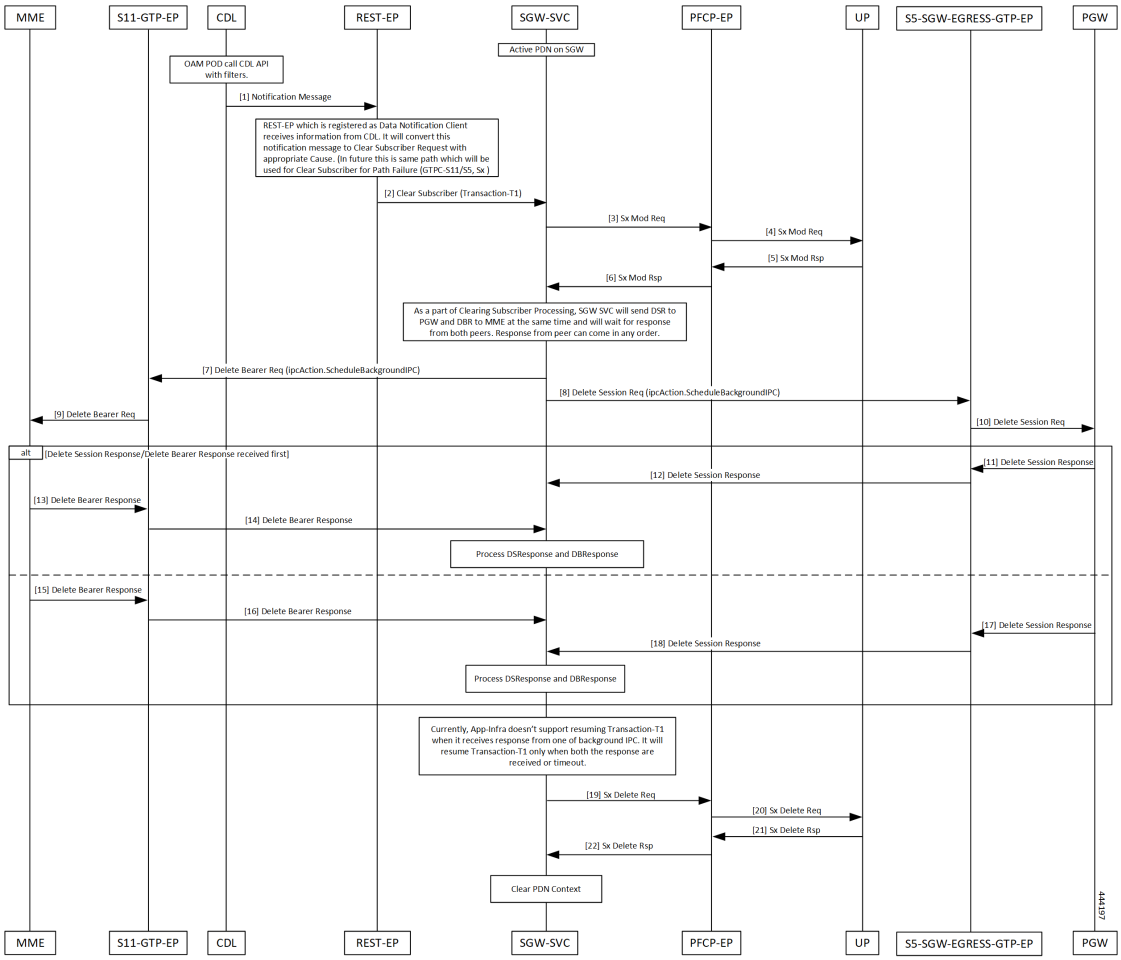
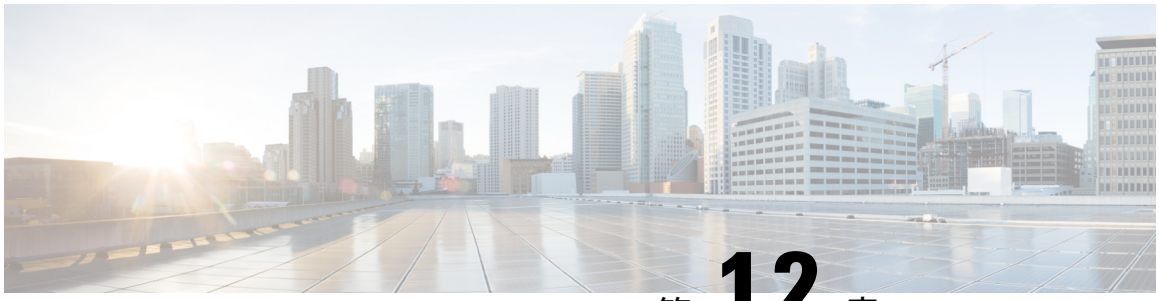


表 *35 :* クリア *PDN* コールフローの説明

ステップ	説明
1	OAM ポッドは、フィルタを使用して CDL API を呼び出します。 CDL は REST EP に通知メッセージを送信します。
2	REST-EP はこのメッセージを原因とともにサブスクライバのクリア要求に変換し、サブスクライバのクリアを SGW サービスポッドに送信します。 トランザクション T1 が開始されます。
3 ～ 6	SGW サービスポッドが PFCP-EP を介して UPF に Sx 変更要求を送信します。 SGW サービスポッドは PFCP-EP を介して UPF から Sx 変更応答を受信します。
7	SGW-Service ポッドが S11-GTP-EP にベアラー削除要求を送信します。

ステップ	説明
8	SGW サービスポッドが S5-SGW-EGRESS-GTP-EP にセッション削除要求を送信します。
9	S11-GTP-EP が MME にベアラー削除要求を送信します。
10 ～ 12	S5-SGW-EGRESS-GTP-EP がセッション削除要求を PGW に送信します。 PGW が S5-SGW-EGRESS-GTP-EP にセッション削除応答を送信します。 S5-SGW-EGRESS-GTP-EP はこの要求を SGW サービスポッドに転送します。
13-16	MME が S11-GTP-EP にベアラー削除応答を送信します。 S11-GTP-EP が SGW サービスポッドに転送します。
17、18	PGW が S5-SGW-EGRESS-GTP-EP にセッション削除応答を送信します。 S5-SGW-EGRESS-GTP-EP はこの要求を SGW サービスポッドに転送します。
19 ～ 22	SGW サービスポッドが Sx 削除要求を PFCP-EP に送信します。 PFCP-EP は要求を UPF に転送します。 UPF は Sx 削除応答を PFCP-EP に送信し、PFCP-EP は SGW サービスポッドに転送します。



第 12 章

コンテキスト置換のサポート

- 機能の概要と変更履歴 （133 ページ）
- 機能説明 （134 ページ）
- 機能の仕組み （134 ページ）
- OAM サポート （140 ページ）

機能の概要と変更履歴

要約データ

表 *36* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *37* : マニュアルの変更履歴

改訂の詳細	リリース
部分的なコンテキスト置換のサポートが導入されました。	2021.02.0
最初の導入。	2020.01.0

機能説明

cnSGW-C は、既存の EBI を使用して Create Session Request（CSReq）を受信する場合、コンテキスト置換をサポートします。MME ノードと cnSGW-C が同期されていない場合、セッションは MME でローカルに終了します。MME は、cnSGW-C にすでに存在する EBI を使用して CSReq を送信します。CSReq に値がゼロ以外の TEID が含まれている場合、cnSGW-C はコンテキストを部分的に置き換えます。TEID がゼロの場合、cnSGW-C は完全なコンテキスト置換を実行します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

完全なコンテキスト置換コールフロー

ここでは、完全なコンテキスト置換コールフローについて説明します。

セッション作成要求のコールフロー

このセクションでは、セッション作成要求のコールフローについて説明します。

図 *21* : セッション作成要求（コンテキスト置換 - シングルまたはマルチ *PDN* サブスクライバ）コールフロー

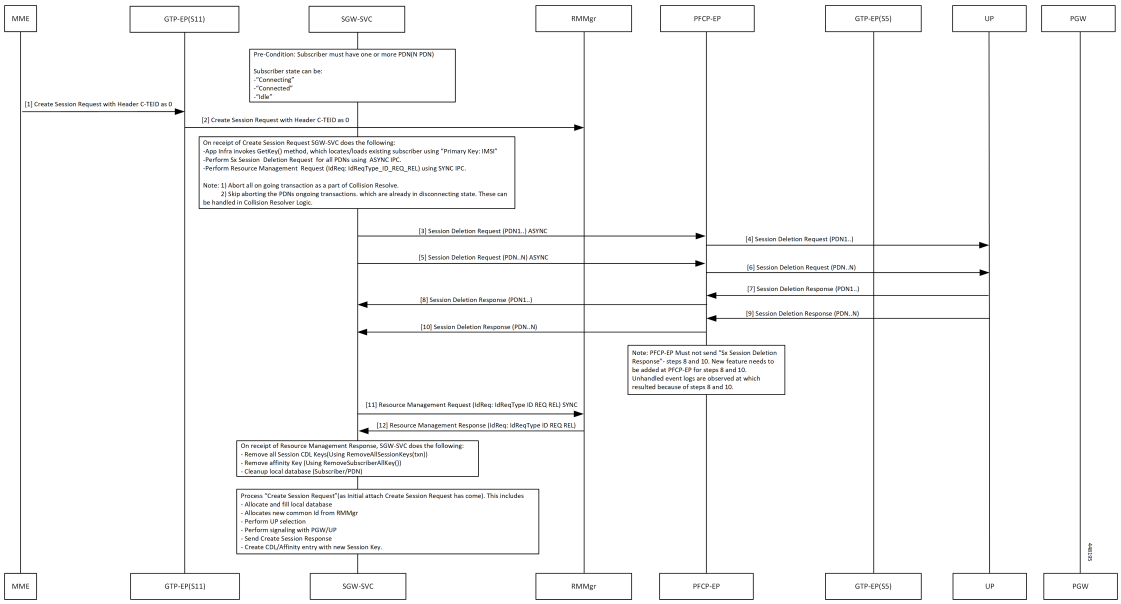


表 *38 :* セッション作成要求（コンテキスト置換 - シングルまたはマルチ *PDN* サブスクライバ）コールフローの説明

ステップ	説明
1	MME が、C-TEID がゼロのセッション作成要求を GTPC-EP 入力に送信します。
2	GTPC-EP 入力が、セッション作成要求を SGW-SVC に転送します。 次の処理が行われます。 • App Infra が GetKey() メソッドを呼び出し、プライマリキー（IMSI）を使用して既存のサブスクライバを見つけてロードします。 • 非同期 IPC を使用してすべての PDN の Sx セッション削除要求を実行します。 • SYNC IPC を使用してリソース管理要求（IdReq：IdReqType_ID_REQ_REL）を実行します。
3、5	SGW サービスポッドが、PDN 1 - N のセッション削除要求を PFCP-EP に送信します。
4、6	PFCP-EP が、PDN 1 - N のセッション削除要求を UPF に転送します。
7、9	PFCP-EP が UPF から PDN 1 から N のセッション削除応答を受信します。
8、10	PFCP-EP が、PDN 1 - N のセッション削除応答を SGW サービスポッドに転送します。
11	SGW サービスポッドが、要求 ID タイプを要求 REL として RMMgr にリソース管理要求を送信します。
12	SGW サービスポッドが、RMMgr からリソース管理応答を要求 ID タイプ（要求 REL）とともに受信します。 SGW サービスポッドでは、次の機能が実行されます。 • すべてのセッション CDL キーの削除（RemoveAllSessionKeys(txn) を使用） • アフィニティキーの削除（RemoveSubscriberAllKey() を使用） • ローカルデータベース（サブスクライバ/PDN）のクリーンアップ

（注） UPF からの削除応答の未処理イベントは無視できます。

部分コンテキスト置換のコールフロー

ここでは、部分コンテキスト置換のコールフローについて説明します。

cnSGW-C が既存の EBI と TEID がゼロ以外である CSReq を受信すると、cnSGW-C は次のコールフローを呼び出すことによって部分的なコンテキスト置換を実行します。

- CSReq で受信した EBI は、既存のデフォルトベアラー用です。
- CSReq で受信した EBI は、既存の専用ベアラー用です。

デフォルトのベアラー EBI コールフローを使用したセッション作成要求

このセクションでは、デフォルトのベアラーEBIコールフローを使用したセッション作成要求について説明します。

図 *22*：デフォルトのベアラー *EBI* を使用した *CSReq* コールフロー

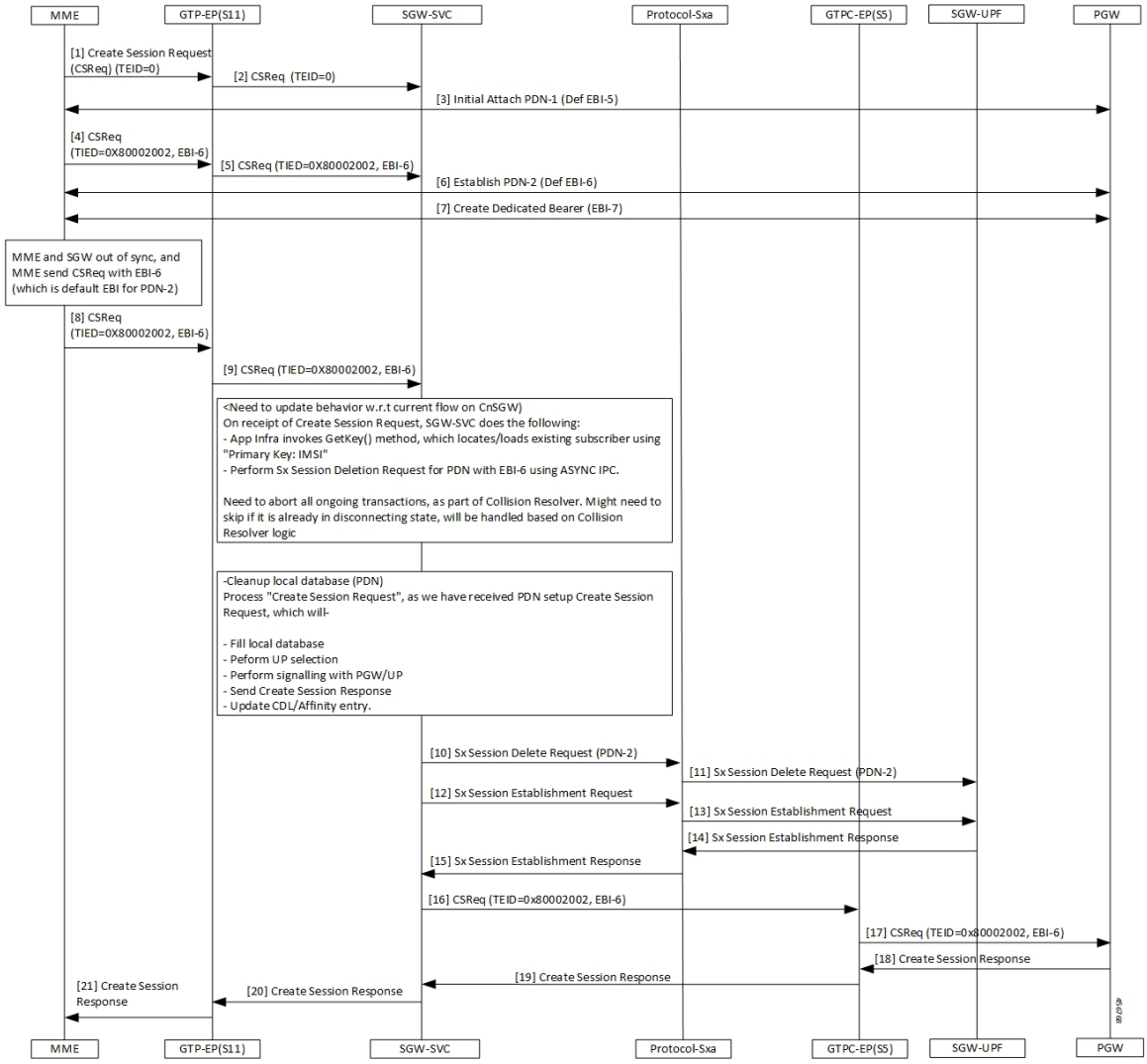


表 *39*：デフォルトのベアラー *EBI* を使用した *CSReq* コールフローの説明

ステップ	説明
1	MME が TIED 値がゼロのセッション作成要求を GTPC-EP（S11）に送信します。

ステップ	説明
2	GTPC-EP（S11）が TIED 値が 0 の Create Session Request を SGW-SVC に転送します。
3	MME と PGW がデフォルトの EBI-5 プロセスを使用して初期接続 PDN-1 を処理します。
4	MME が TIED=0x80002002、EBI-6 を含むセッション作成リクエストを GTPC-EP に送信します。
5	GTPC-EP が TIED=0x80002002、EBI-6 を含むセッション作成リクエストを SGW-SVC に転送します。
6	MME と PGW が、デフォルトの EBI-6 接続を使用して PDN-2 を確立します。
7	MME と PGW が EBI-7 プロセスを使用して専用ベアラー作成を完了します。
8	SGW と MME が同期していない場合、MME が SGW 内に存在する EBI-6 を使用してセッション作成要求を送信します。
9	GTPC-EP が CSReq TIED= 0x80002002 と EBI-6 を SGW に送信します。
10	セッション作成要求を受信すると、SGW-SVC は次の処理を実行します。 • デフォルトの EBI=6 で PDN をクリーンアップします。 • Sx シグナリングを UPF に送信してセッションをクリアします。 • 新しい PDN セットアップとしてセッション作成要求を実行します。 SGW が PDN-2 の Sx セッション削除応答を Protocol-SXA に送信します。
11	Protocol-SXA が Sx セッション削除要求を SGW-UPF に転送します。
12	SGW がセッション確立要求を Protocol-SXA に送信します。
13	Protocol-SXA が Sx セッション確立要求を SGW-UPF に転送します。
14	SGW-UPF が Sx セッション確立応答で Protocol-SXA に応答します。
15	Protocol-SXA は SGW-SVC に Sx セッション確立応答を送信します。
16	SGW-SVC が TIED=0x80002002、EBI-6 を含むセッション作成リクエストを GTPC-EP に送信します。
17	GTPC-EP が TIED=0x80002002、EBI-6 を含むセッション作成リクエストを PGW に送信します。
18	PGW がセッション作成応答を GTPC-EP に送信します。
19	GTPC-EP がセッション作成応答で SGW-SVC に応答します。

ステップ	説明
20	SGW-SVC がその応答を GTPC-EP に転送します。
21	GTPC-EP がセッション作成応答を MME に送信します。

専用ベアラー EBI を使用した Create Session Request コールフロー

ここでは、専用ベアラー EBI を使用した Create Session Request コールフローについて説明します。

図 *23 :* 専用ベアラー *EBI* を使用した *CSReq* コールフロー

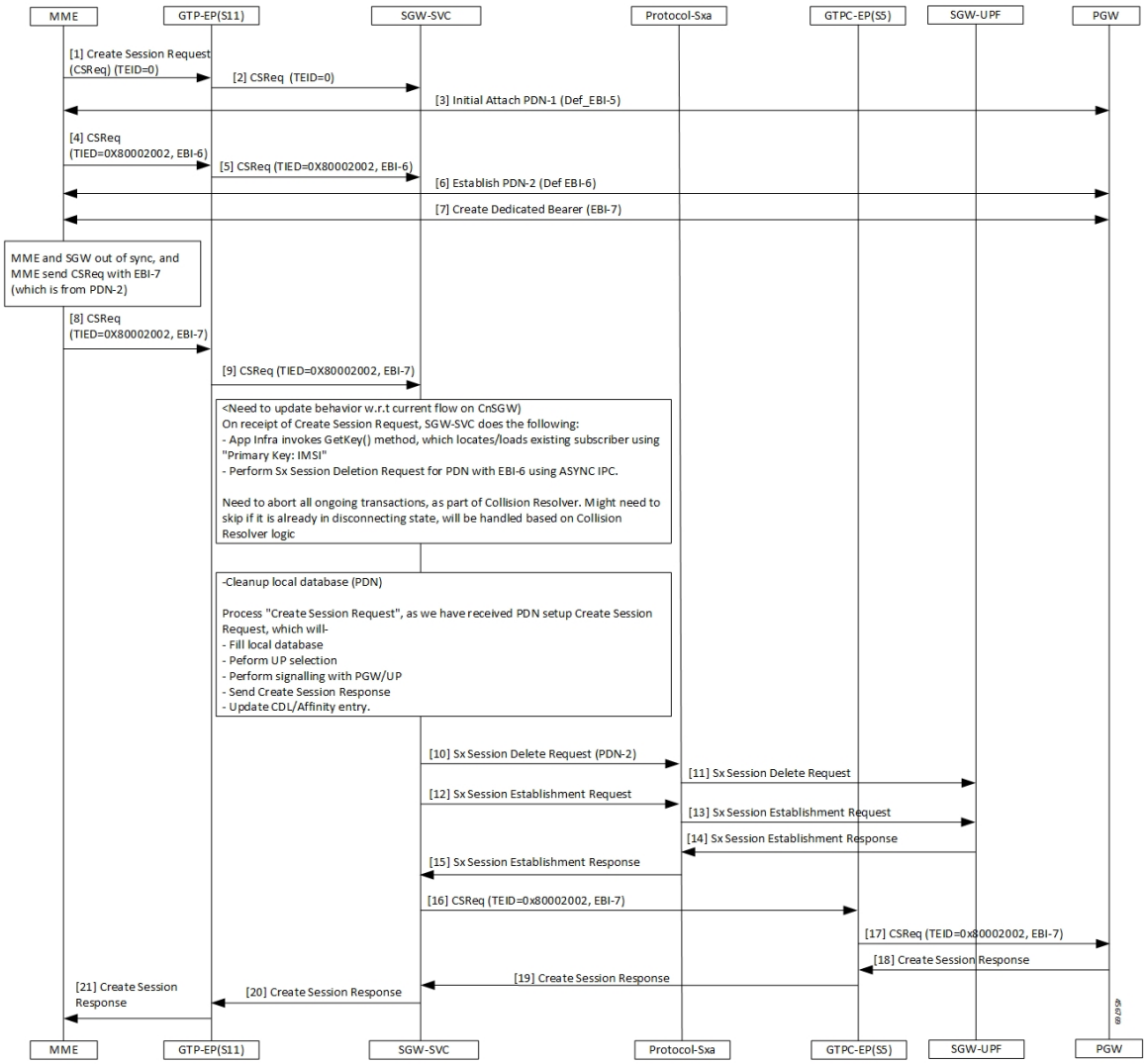


表 *40 :* 専用ベアラー *EBI* を使用した *CSReq* コールフローの説明

ステップ	説明
1	MME が TIED 値がゼロの Create Session Request を GTPC-EP（S11）に送信します。
2	GTPC-EP（S11）が TIED 値がゼロの Create Session Request を SGW-SVC に転送します。
3	MME と PGW が EBI-5 プロセスを使用して初期接続 PDN を処理します。
4	MME が EBI-6 を使用して Create Session Request を GTPC-EP に送信します。
5	GTPC-EP が EBI-6 を使用して Create Session Request を SGW-SVC に転送します。
6	MME と PGW が EBI-6 接続を使用して PDN を確立します。
7	MME と PGW が EBI-7 プロセスを使用して Create Dedicated Bearer を完了します。
8	SGW と MME が同期していない場合、MME が SGW 内に存在する EBI-6 を使用して Create Session Request を送信します。
9	GTPC-EP が EBI-7 を使用して CSReq を SGW に送信します。
10	Create Session Request を受信後、SGW-SVC が以下を実行します。 • デフォルトの EBI=6 を使用して PDN をクリーンアップします。 • Sx シグナリングを UPF に送信してセッションをクリアします。 • 新しい PDN セットアップとして Create Session Request を実行します。 SGW が PDN-2 の Sx Session Deletion Request を Protocol-SXA に送信します。
11	Protocol-SXA が Sx Session Deletion Request を SGW-UPF に転送します。
12	SGW が Session Establishment Request を Protocol-SXA に送信します。
13	Protocol-SXA が Sx Session Establishment Request を SGW-UPF に転送します。
14	SGW-UPF が Sx Session Establishment Response で Protocol-SXA に応答します。
15	Protocol-SXA が Sx Session Establishment Response を SGW-SVC に送信します。
16	SGW-SVC が TIED=0x80002002、EBI-7 を含む Create Session Request を GTPC-EP に送信します。
17	GTPC-EP が TIED=0x80002002、EBI-7 を含む Create Session Request を PGW に送信します。
18	PGW が Create Session Response を GTPC-EP に送信します。

ステップ	説明
19	GTPC-EP が Create Session Response で SGW-SVC に応答します。
20	SGW-SVC がその応答を GTPC-EP に転送します。
21	GTPC-EP が Create Session Response を MME に送信します。

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートに関して説明します。

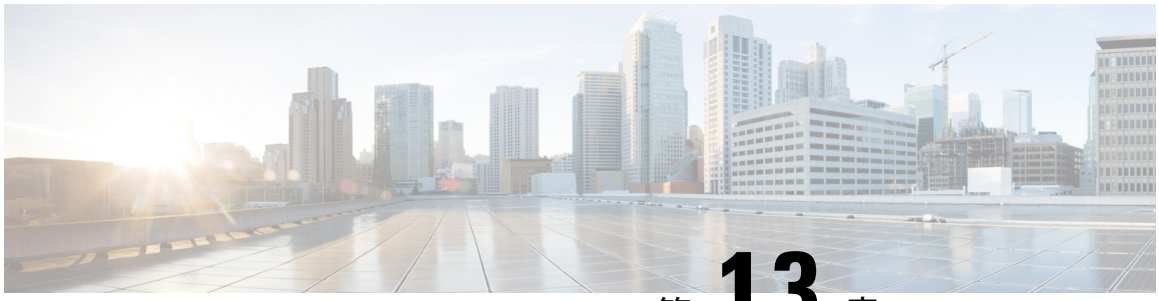
バルク統計

部分的なコンテキスト置換機能では、次の統計情報がサポートされています。

- sgw_pdn_disconnect_stats：切断ステータスの SGW PDN の総数をキャプチャします。

  Prometheus クエリの例：

```
sgw_pdn_disconnect_stats{app_name="smf",cluster="cn",data_center=\
"cn",instance_id="0",pdn_type="ipv4",rat_type="EUTRAN",reason="context_replacement",\
 service_name="sgw-service"} 1
```



第 13 章

専用ベアラーのサポート

- 機能の概要と変更履歴 （141 ページ）
- 機能説明 （142 ページ）
- 専用ベアラーの構成と更新 （142 ページ）
- 専用ベアラーの削除 （149 ページ）

機能の概要と変更履歴

要約データ

表 *41* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *42* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

専用ベアラーの構成と更新

cnSGW-C は、単一/複数の専用ベアラーの作成と更新をサポートしています。

専用ベアラーの削除

cnSGW-C は、単一/複数の専用ベアラーの削除をサポートしています。

専用ベアラーの構成と更新

機能説明

cnSGW-Cは、単一のPDNサブスクライバと複数のPDNサブスクライバの両方の専用ベアラーの作成と更新をサポートします。また、単一のベアラー作成手順の一部として複数ベアラーコンテキストもサポートしています。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

専用ベアラーのセットアップ：要求承認済みコールフロー

このセクションでは、専用ベアラーのセットアップ：要求承認済みコールフローについて説明します。

図 *24 :* 専用ベアラーのセットアップ：要求承認済みコールフロー

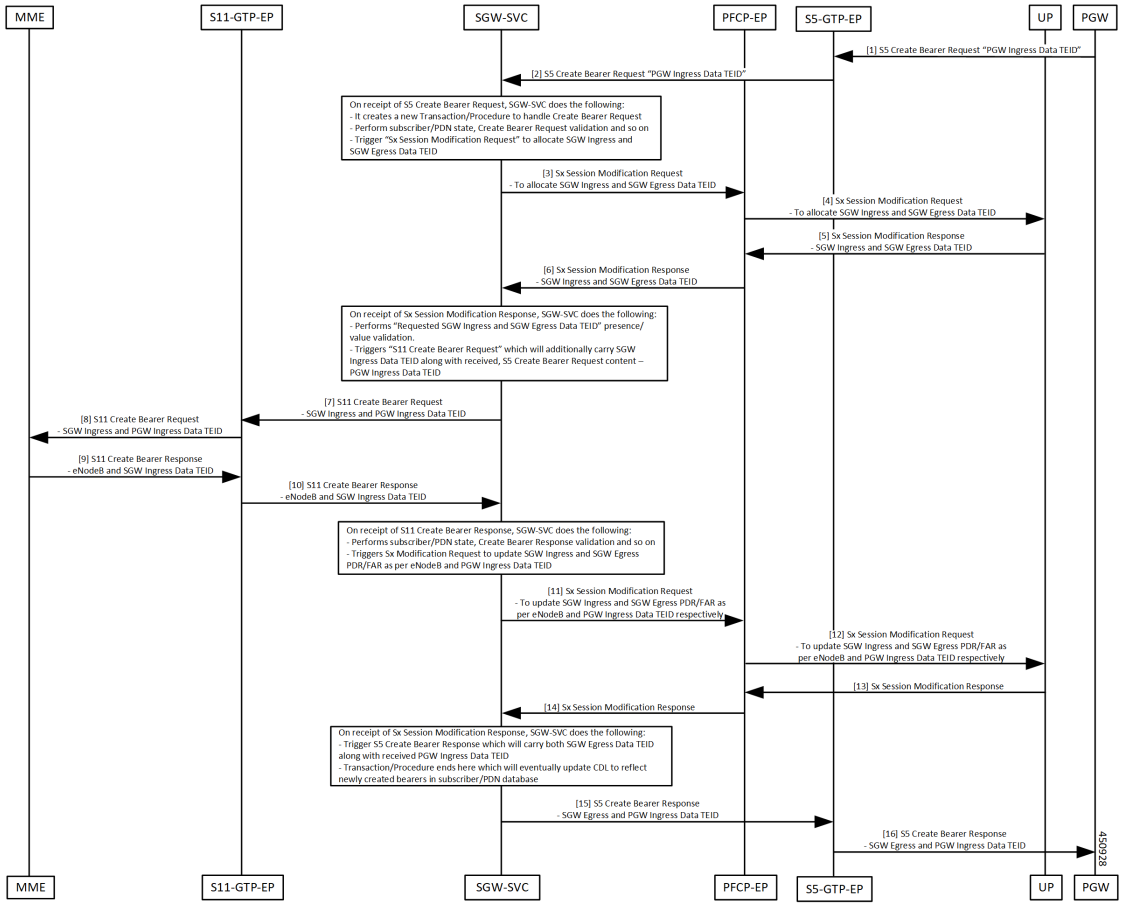


表 *43 :* 専用ベアラーのセットアップ：要求承認済みコールフローの説明

ステップ	説明
1	PGW が S5 Create Bearer Request を S5-GTPC-EP ポッドに送信します。
2	S5-GTPC-EP ポッドが S5 ベアラー作成要求を SGW-SVC ポッドに転送します。
3	SGW-SVC が S5 ベアラー作成要求を受信し、次の操作を実行します。 • 新しいトランザクションの作成 • GTP 検証の実行 • PFCP-EP ポッドへの Sx 変更要求のトリガー
4	PFCP-EP ポッドが、SGW 入力および SGW 出力 TEID を割り当てるために、Sx 変更要求を UP に転送します。

ステップ	説明
5	PFCP-EP ポッドが、SGW 入力および SGW 出力 TEID を含む Sx 変更応答を UP から受信します。
6	PFCP-EP ポッドが、SGW 入力および SGW 出力 TEID を含む Sx 変更応答を SGW-SVC に転送します。
7	SGW-SVC が Sx 変更要求を受信し、次を実行します。 • 受信した SGW 入力および SGW 出力 TEID の検証 • S11-GTPC-EP ポッドへの SGW 入力 TEID を使用した S11 ベアラー要求のトリガー
8	S11-GTPC-EP ポッドが、SGW 入力 TEID を持つ S11 ベアラー作成要求を MME に転送します。
9	MME が S11 ベアラー作成応答を S11-GTPC-EP ポッドに送信します。
10	S11-GTPC-EP ポッドが、S11 ベアラー作成応答を SGW-SVC に転送します。
11	SGW-SVC が S11 ベアラー作成応答を受信し、次を実行します。 • GTP の検証 • PFCP-EP ポッドへの Sx 変更要求をトリガーし、MME および PGW GTPU-TEID を使用して SGW 入力および SGW 出力 PDR/FAR を更新
12	PFCP-EP ポッドが Sx 変更要求を UP に転送します。
13	UP が Sx 変更要求を PFCP-EP ポッドに送信します。
14	PFCP-EP ポッドが Sx 変更応答を SGW-SVC ポッドに転送します。
15、16	SGW-SVC ポッドが Sx 変更応答を受信し、次を実行します。 • トランザクション/手順の終了 • CDL の更新 • 一致する PGW GTPU TEID を持つ SGW 出力 TEID を含む S5 ベアラー作成応答を送信します。原因は Accepted です。

専用ベアラーのセットアップ：要求の一部承認コールフロー

ここでは、専用ベアラーのセットアップコールフローについて説明します。この手順では、MME が Request Accepted Partially の理由で GTP とともに Create Bearer Response を送信します。

前提条件：2 つのベアラーコンテキストを使用してベアラー手順を作成します。

図 *25 :* 専用ベアラーのセットアップ：要求の一部承認コールフロー

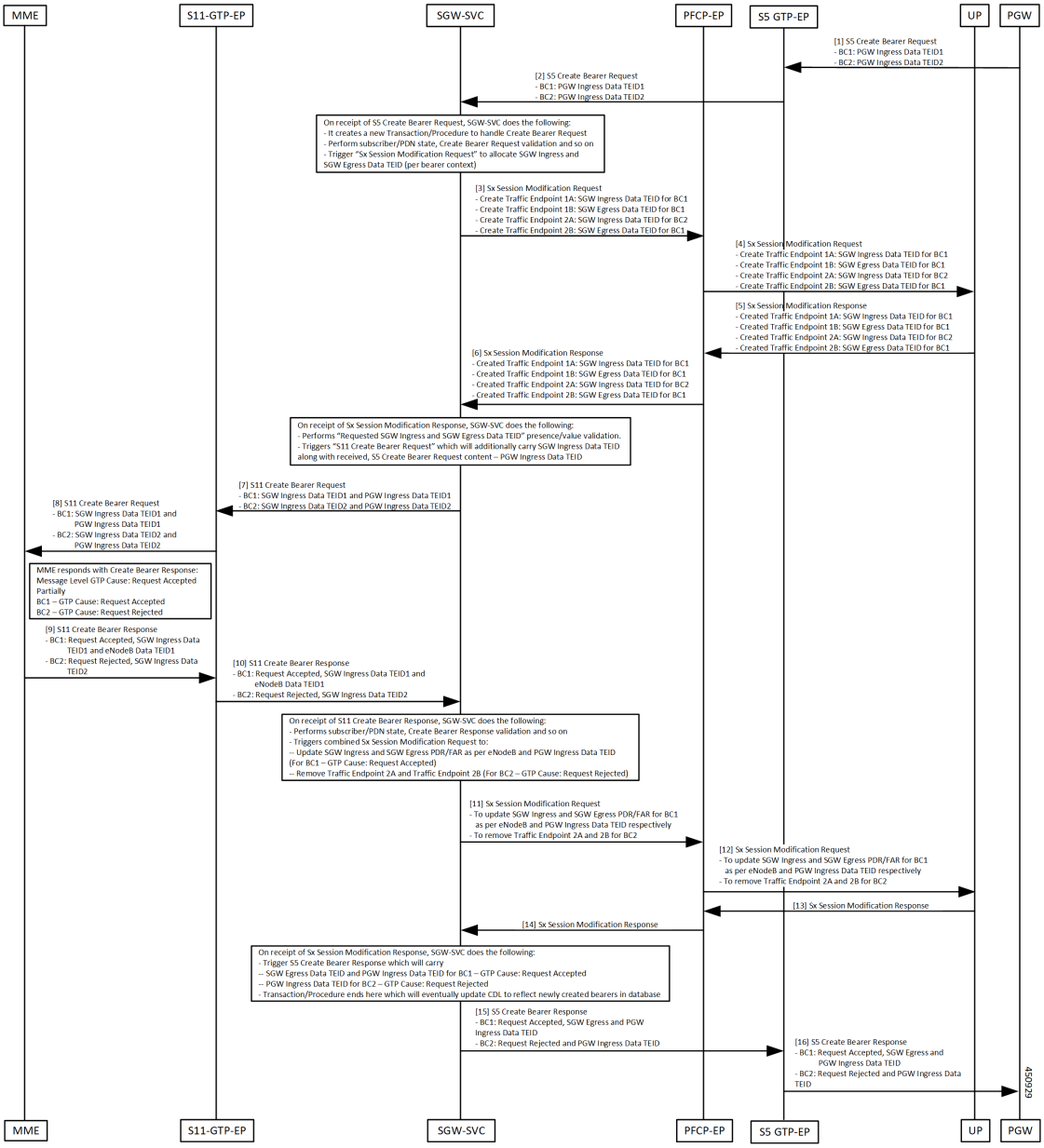


表 *44 :* 専用ベアラーのセットアップ：要求の一部承認コールフローの説明

ステップ	説明
1	PGW が複数のベアラーコンテキストとともに S5 Create Bearer Request を S5-GTPC-EP ポッドに送信します。
2	S5-GTPC-EP ポッドが S5 Create Bearer Request を SGW-SVC ポッドに転送します。

ステップ	説明
3	SGW-SVC ポッドが S5 Create Bearer Request を受信し、次を実行します。 • 新しいトランザクションを作成します。 • GTP 検証を実行します。 • Sx Modification Request をトリガーして、SGW 入力および SGW 出力 TEID を PFCP-EP ポッドに割り当てます。
4	PFCP-EP ポッドが Sx Modification Request を UP に転送します。
5	UP が Sx Modification Response を PFCP-EP ポッドに送信します。
6	PFCP-EP ポッドが Sx Modification Response を SGW-SVC ポッドに転送します。
7	SGW-SVC が Sx Modification Response を受信し、次を実行します。 • 受信した SGW 入力および SGW 出力 TEID を検証します。 • SGW 入力 TEID とともに S11 Create Bearer Request を S11-GTPC-EP ポッドにトリガーします。
8	S11-GTPC-EP ポッドが S11 Create Bearer Request を MME に転送します。
9	S11-GTPC-EP が MME から S11 Create Bearer Response を受信します。メッセージレベル GTP の理由は Request Accepted Partially です。 • 一部のベアラーコンテキストの場合、GTP の理由は Request Accepted です • 一部のベアラーコンテキストの場合、GTP の理由は Request Rejected です
10	S11-GTPC-EP ポッドが S11 Create Bearer Response を SGW-SVC ポッドに転送します。
11	SGW-SVC ポッドが S11 Create Bearer Response を受信し、次を実行します。 • GTP 検証 • 成功したベアラーの場合：MME および PGW GTPU-TEID を使用して SGW 入力および SGW 出力 PDR/FAR を更新するために、PFCP-EP ポッドに Sx Modification Request をトリガーします。 • 失敗したベアラーの場合：トラフィックエンドポイントを削除します。
12	PFCP-EP ポッドが Sx Modification Request を UP に転送します。
13	UP が Sx Modification Response を PFCP-EP ポッドに送信します。
14	PFCP-EP ポッドが Sx Modification Response を SGW-SVC ポッドに転送します。

ステップ	説明
15、16	SGW-SVC ポッドが Sx Modification Response を受信し、次を実行します。 • トランザクション/手順の終了 • CDL の更新 • 成功したベアラーの場合：一致する PGW GTPU TEID を使用して SGW 出力 TEID を含む S5 Create Bearer Response を送信します。 • 失敗したベアラーの場合：メッセージレベルの理由を Partially Accepted として、ベアラーのコンテキストをそのまま送信します。

専用ベアラーの更新：要求承認コールフロー

このセクションでは、デフォルト/専用ベアラー更新手順のコールフローについて説明します。

単一のベアラー更新手順は、以下をサポートしています。

- デフォルトベアラー QoS/TFT の変更
- 単一/複数の専用ベアラー QoS/TFT の変更
- APN-AMBR の変更

（注） コールフローには、デフォルト/専用ベアラーの更新手順に関連する Sx 通信メッセージは含まれていません。

図 *26 :* デフォルト/専用ベアラー更新（単一/複数ベアラー）のサポートコールフロー

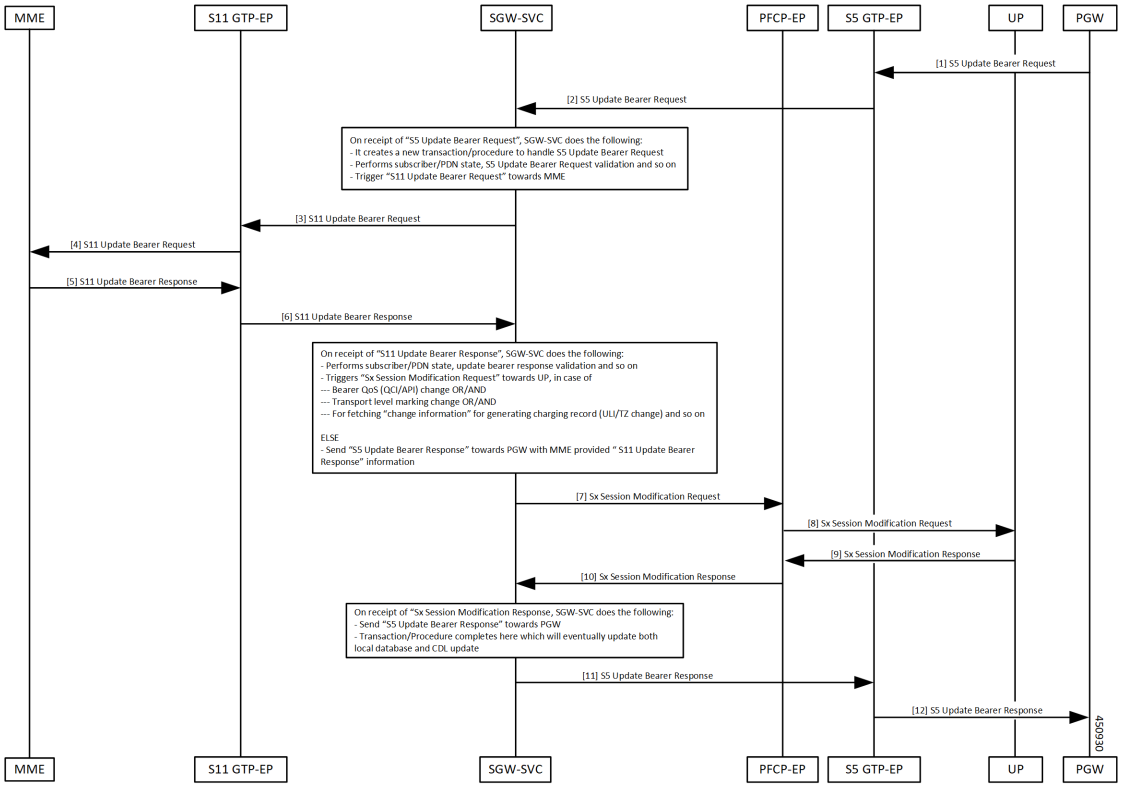


表 *45 :* デフォルト/専用ベアラー更新（単一/複数ベアラー）のサポートコールフローの説明

ステップ	説明
1	PGW が複数のベアラーコンテキストとともに S5 ベアラー更新リクエストを GTPC-EP ポッドに送信します。
2	GTPC-EP ポッドは、S5 ベアラー更新要求を SGW-SVC ポッドに転送します。
3	SGW-SVC は S5 ベアラー更新要求を受信し、次を実行します。 • 新しいトランザクションの作成 • GTP 検証の実行 • GTPC-EP ポッドへの S11 ベアラー更新要求のトリガー
4	GTPC-EP ポッドが、S11 ベアラー更新要求を MME に転送します。
5	MME が S11 ベアラー更新応答を GTPC-EP ポッドに送信します。
6	GTPC-EP ポッドが、S11 ベアラー更新応答を SGW-SVC ポッドに転送します。

ステップ	説明
7	SGW-SVC が S11 ベアラー更新応答を受信し、GTP 検証を実行します。 • 次のいずれかが当てはまる場合、SGW-SVC は PFCP-EP ポッドに対する Sx 変更要求をトリガーします。 • ベアラー QoS（QCI/ARP）の変更 • トランスポート レベル マーキングの変更 • ULI/TZ の変更に伴う課金レコードを生成するための課金情報の取得 • 他の場合：SGW-SVC は、MME から提供された S11 ベアラー更新応答の情報を含めて、S11 ベアラー更新応答を PGW に送信します。
8	PFCP-EP ポッドが Sx セッション変更要求を UP に転送します。
9	UP が Sx セッション変更要求を PFCP-EP ポッドに送信します。
10	PFCP-EP ポッドが Sx セッション変更応答を SGW-SVC に転送します。 SGW-SVC が Sx 変更要求を受信し、次を実行します。 • トランザクション/手順の終了 • CDL の更新 • 原因を Accepted として、S5 ベアラー更新応答を送信
11	SGW-SVC が、S5 ベアラー更新応答を PFCP-EP ポッドに送信します。 PFCP-EP ポッドが、S5 ベアラー更新応答を GTP-EP ポッドに転送します。
12	GTP-EP ポッドが、S5 ベアラー更新応答を UP に転送します。 UP が、S5 ベアラー更新応答を PGW に転送します。

専用ベアラーの削除

機能説明

cnSGW-C は、単一のベアラー削除手順の一部として、単一/複数の専用ベアラーの削除をサポートしています。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

専用ベアラー削除手順のコールフロー

このセクションでは、専用ベアラー削除手順のコールフローについて説明します。

図 *27 :* 専用ベアラー削除手順（単一/複数ベアラー）のコールフロー

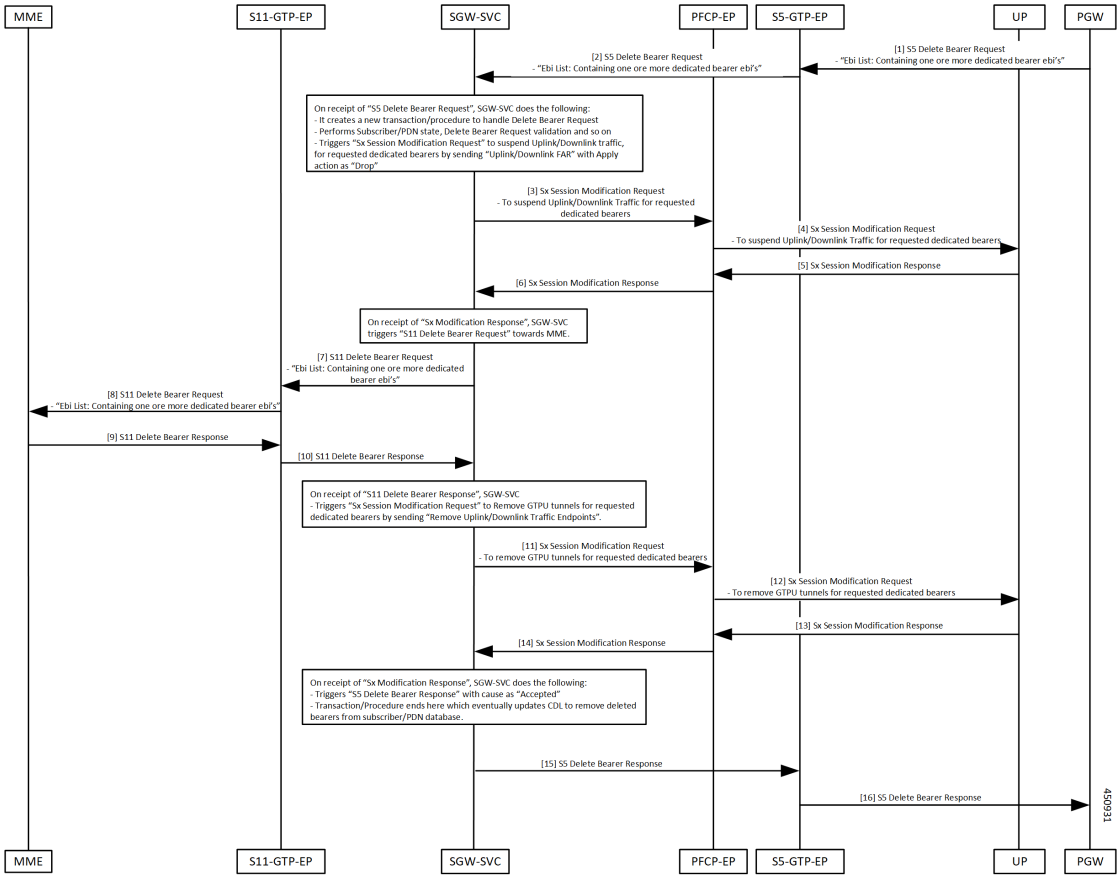
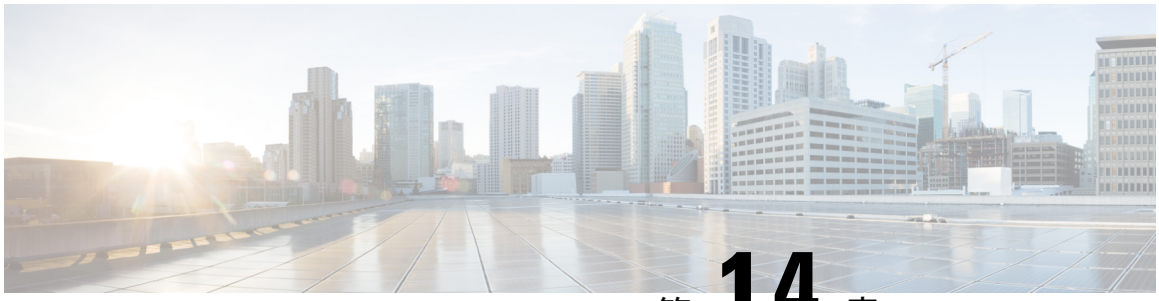


表 *46 :* 専用ベアラー削除手順（単一/複数ベアラー）のコールフローの説明

ステップ	説明
1	PGW は、1 つ以上の専用ベアラー EBI を含む EBI リストを添えて、S5 ベアラー削除要求を GTPC-EP ポッドに送信します。

ステップ	説明
2	GTPC-EP ポッドは、S5 S5 ベアラー削除要求を SGW-SVC ポッドに転送します。
3	SGW-SVC ポッドが S5 ベアラー削除要求を受信し、次を実行します。 • 新しいトランザクションの作成 • GTP 検証の実行 • 要求されたベアラーに対するアップリンク/ダウンリンクトラフィックを一時停止するため、PFCP-EP ポッドに対する Sx 変更要求をトリガー
4	PFCP-EP ポッドが Sx 変更要求を UP に転送します。
5	UP が Sx 変更要求を PFCP-EP ポッドに送信します。
6	PFCP-EP ポッドが Sx 変更応答を SGW-SVC ポッドに転送します。
7	SGW-SVC ポッドが Sx 変更応答を受信し、GTPC-EP ポッドに対する S11 ベアラー削除要求をトリガーします。
8	GTPC-EP ポッドが、S11 ベアラー削除要求を MME に転送します。
9	MME が S11 ベアラー削除応答を GTPC-EP ポッドに送信します。
10	GTPC-EP ポッドが、S11 ベアラー削除応答を SGW-SVC ポッドに転送します。
11	SGW-SVC は S11 ベアラー削除要求を受信すると、PFCP-EP ポッドに対して Sx 変更要求をトリガーし、要求された専用ベアラーの GTPU トンネルを削除するためにトラフィックエンドポイントを削除します。
12	PFCP-EP ポッドが、Sx 変更要求を UP に転送して、要求された専用ベアラーの GTP トンネルを削除します。
13	UP が Sx 変更要求を PFCP-EP ポッドに送信します。
14	PFCP-EP が Sx 変更応答を SGW-SVC ポッドに転送します。
15	SGW-SVC が Sx 変更要求を受信し、次を実行します。 • トランザクション/手順の終了 • CDL の更新 • 原因を Accepted として、S5 ベアラー削除応答を GTP-EP ポッドに送信
16	GTP-EP ポッドは、S5 ベアラー削除要求を PGW に転送します。



第 14 章

削除ベアラーおよび削除セッション要求

- 機能の概要と変更履歴 （153 ページ）
- 機能説明 （153 ページ）
- 機能の仕組み （155 ページ）

機能の概要と変更履歴

要約データ

表 *47* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *48* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2020.04

機能説明

この機能は次をサポートします。

- MME からのセッションの削除要求
- PGW からのベアラーの削除要求

この削除は SGW での PDN 接続をクリアすることに役立ち、それにより cnSGW-C のリソースがクリアされ、関連するすべての TEID が解放されます。

MME から削除

1. cnSGW-C が Sx 変更要求をユーザープレーン（UP）に送信し、転送アクションを DROP としてマークします。これにより、すべてのアップリンクパケットとダウンリンクパケットが SGW-U でドロップされるようにします。
2. cnSGW-C は PGW/SMF にセッション削除要求を送信します。
3. SGW が PGW/SMF からセッションの削除応答を受信すると、cnSGW-C は Sx 終了要求を UP に送信してセッションをクリアします。
4. UP が SGW-U セッションの削除を確認すると、cnSGW-C はノードマネージャに要求を送信し、MME にセッション削除応答を送信して、割り当てられた ID を解放します。

PGW から削除

1. cnSGW-C が Sx 変更要求を UP に送信し、転送アクションを DROP としてマークします。これにより、すべてのアップリンクパケットとダウンリンクパケットが SGW-U でドロップされるようにします。
2. cnSGW-C が、MME にベアラーの削除要求を送信します。
3. SGW が MME からベアラー削除応答を受信すると、cnSGW-C は Sx 終了要求を UP に送信してセッションをクリアします。
4. UP が SGW-U セッションの削除を確認すると、cnSGW-C はノードマネージャに要求を送信し、PGW にベアラー削除応答を送信して、割り当てられた ID を解放します。

標準規格への準拠

ベアラー削除およびセッション削除要求のサポート機能は、次の標準規格に準拠しています。

- 3GPP TS 23.401「General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access」
- 3GPP TS 23.214「Architecture enhancements for control and user plane separation of EPC nodes」
- *3GPP TS 29.274 "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3"*
- 3GPP TS 29.244「Interface between the Control Plane and the User Plane nodes」

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

図 *28 : MME* からの削除コールフロー

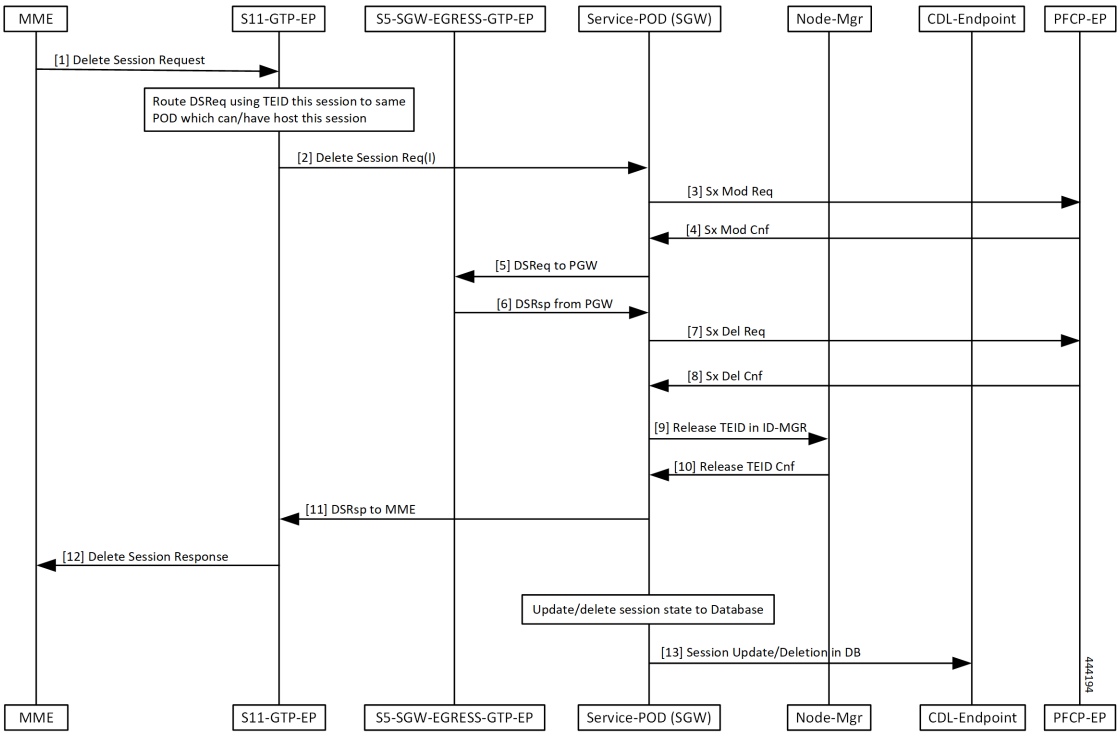


表 *49 : MME* からの削除コールフローの説明

ステップ	説明
1	MME が Delete Session Request を S11-GTP-EP に送信します。
2	S11-GTP-EP が、このセッションを処理するサービスポッド（SGW）に、TEID 値とともにこのメッセージをルーティングします。
3	サービスポッド（SGW）が Sx Modification Request を PFCP-EP に送信します。
4	PFCP-EP が Sx Modification Confirmation をサービスポッド（SGW）に送信します。

ステップ	説明
5	サービスポッド（SGW）が S5-SGW-EGRESS-GTP-EP を介して Delete Session Request を PGW に送信します。
6	サービスポッド（SGW）が S5-SGW-EGRESS-GTP-EP を介して PGW から Delete Session Request を受信します。
7	サービスポッド（SGW）が Sx Delete Request を PFCP-EP に送信します。
8	サービスポッド（SGW）が PFCP-EP から Sx Delete Confirmation を受信します。
9	サービスポッド（SGW）が ID-MGR の Release TEID を Node-Mgr に送信します。
10	サービスポッド（SGW）が Node-Mgr から Release TEID Confirmation を受信します。
11	サービスポッド（SGW）が Delete Session Response を S11-GTP-EP に送信します。
12	S11-GTP-EP が Delete Session Response を MME に送信します。
13	サービスポッド（SGW）がデータベースメッセージで Session Update または Delete を CDL に送信します。

図 *29 : PGW* からの削除コールフロー

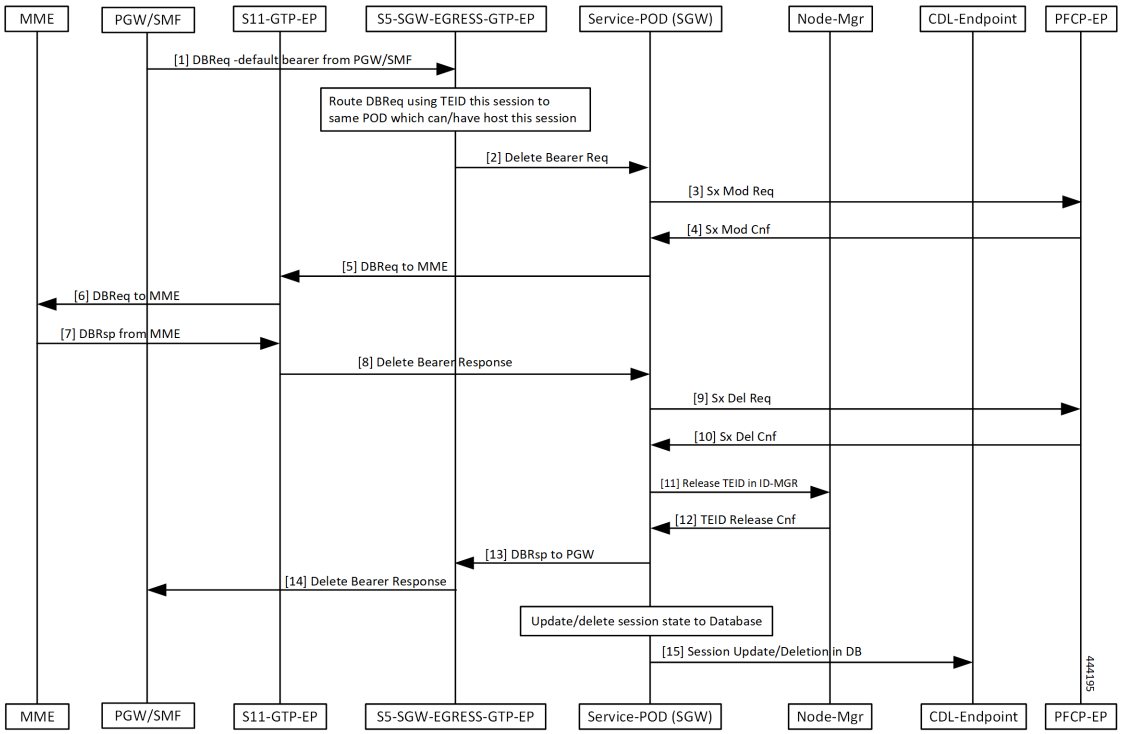
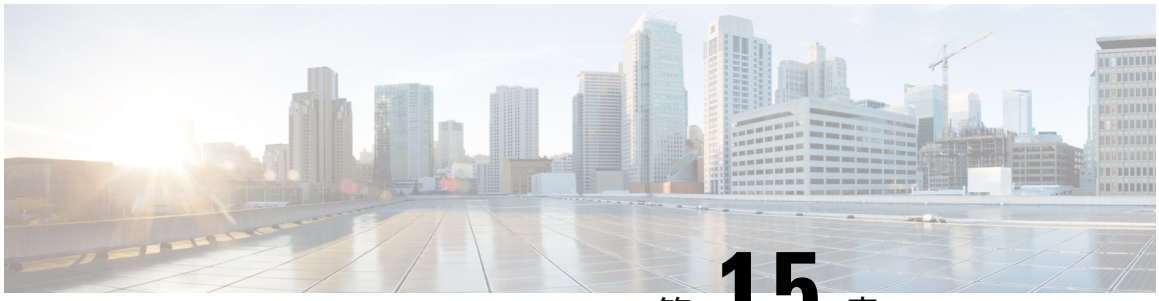


表 *50 : PGW* からの削除コールフローの説明

ステップ	説明
1	PGW/SMF が Delete Bearer Request を S5-SGW-EGRESS-GTP-EP に送信します。
2	S5-SGW-EGRESS-GTP-EP がこのセッションをホストした同じポッドに、TEID 値とともにこのメッセージをルーティングします。 S5-SGW-EGRESS-GTP-EP が Delete Bearer Request をサービスポッド（SGW）に送信します。
3	サービスポッド（SGW）が Sx Modification Request を PFCP-EP に送信し、Sx Modification Confirmation を受信します。
4	PFCP-EP が Sx Modification Confirmation をサービスポッド（SGW）に送信します。
5	サービスポッド（SGW）が S11-GTP-EP を介して Delete Bearer Request を MME に送信します。
6	S11-GTP-EP が Delete Bearer Request を MME に転送します。
7	MME が Delete Bearer Response を S11-GTP-EP に送信します。
8	S11-GTP-EP が Delete Bearer Response をサービスポッド（SGW）に転送します。
9	サービスポッド（SGW）が Sx Delete Request を PFCP-EP に送信します。
10	サービスポッド（SGW）が PFCP-EP から Sx Delete Confirmation を受信します。
11	サービスポッド（SGW）が ID-MGR の Release TEID を Node-Mgr に送信します。
12	サービスポッド（SGW）が Node-Mgr から Release TEID Confirmation を受信します。
13	S11-GTP-EP が S5-SGW-EGRESS-GTP-EP を介して Delete Bearer Response を PGW に送信します。
14	S5-SGW-EGRESS-GTP-EP が Delete Bearer Response を PGW/SMF に送信します。
15	サービスポッド（SGW）がデータベースメッセージで Session Update または Delete を CDL に送信します。



第 15 章

ダウンリンクデータ通知

- 機能の概要と変更履歴 （159 ページ）
- 機能説明 （160 ページ）
- DDN メッセージ処理 （160 ページ）
- 制御メッセージでトリガされる DDN のサポート （169 ページ）
- DDN Advance 機能 （172 ページ）

機能の概要と変更履歴

要約データ

表 *51* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	DDN メッセージ処理のサポート：有効：常時接続 制御メッセージでトリガーされる DDN のサポート：無効：有効にするには設定が必要です DDN の高度な機能：有効：常時接続
関連資料	該当なし

マニュアルの変更履歴

表 *52 :* マニュアルの変更履歴

改訂の詳細	リリース
拡張機能が導入されました。 DDN Advance 機能のサポートが追加されました。	2021.02.0
最初の導入。	2021.01.0

機能説明

次のサブ機能がこの機能に関連付けられています。

- DDN メッセージ処理
- トリガーされる DDN の制御メッセージ
- Downlink Data Notification Delay
- 高優先順位 Downlink Data Notification
- DDN スロットリング

DDN メッセージ処理

機能説明

cnSGW-C は、以下を含む Downlink Data Notification（DDN）機能の処理をサポートしています。

- UE が IDLE 状態の場合、ダウンリンクデータの到着時に UE をページングするための、MME に対する DDN メッセージの生成。
- DDN ACK/DDN Failure Indication の処理。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

ダウンリンクデータ通知成功コールフロー

ここでは、ダウンリンクデータ通知成功コールフローについて説明します。

図 *30* : ダウンリンクデータ通知成功手順のコールフロー

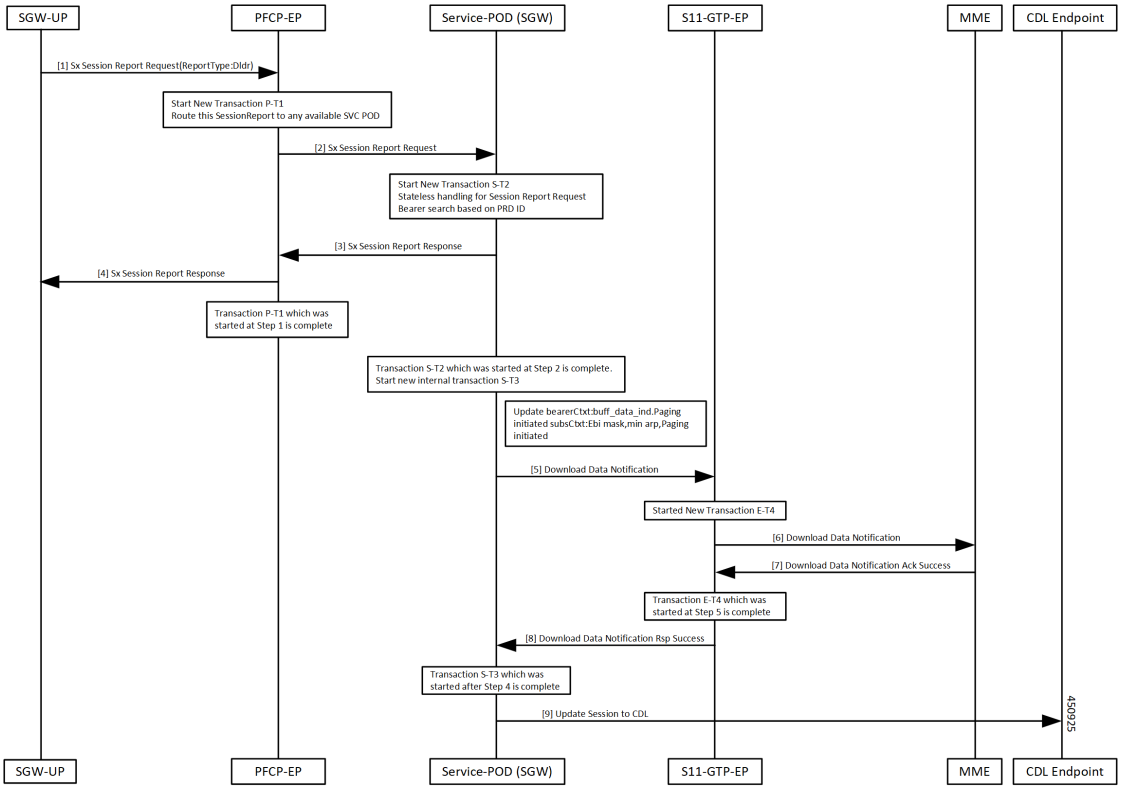


表 *53* : ダウンリンクデータ通知成功手順のコールフローの説明

ステップ	説明
1	SGW-UP が PFCP-EP ポッドにセッションレポート要求を送信します。
2	セッションレポート要求を受信した後、PFCP-EP が次の手順を実行します。 • 新しい P-T1 トランザクションを開始します。 • インターフェイスタイプをチェックします。Sxaインターフェイスの場合、使用可能なSGWサービスポッドを探し、それに応じて要求をルーティングします。 • Sx セッションレポート要求を SGW サービスポッドに送信します。

ステップ	説明
3	Sxセッションレポート要求を受信したら、SGWサービスポッドが次の手順を実行します。 • 新しい S-T2 トランザクションを作成します。 • 受信したメッセージタイプに基づいて、このメッセージに不要な状態処理を更新します。 • 状態処理以外のトランザクションを処理します（このようなメッセージの処理は優先順位が高くなります）。 • PDR IDに基づいてベアラーを検索します。ベアラーが見つからない場合、SGWサービスポッドは、セッションレポート応答で原因を**要求が拒否された**と入力します。 • 要求で受信したレポートタイプが有効でない/サポートされていない場合、SGWサービスポッドは原因を**要求が拒否された**と入力し、Sx セッションレポート応答を送信します。
4	PFCP-EP が SGW-UP に Sx セッションレポート応答を転送します。
5	ステップ 1 で開始された P-T1 トランザクションが完了しました。 SGW サービスポッド側： • ステップ 2 で開始された S-T2 トランザクションが完了しました。 • Sxセッションレポート応答が成功した場合、セッションレポート要求と同じバッファを使用して新しい内部トランザクション S-T3 が開始されます。 • DLDR レポートタイプの DDN 手順が開始されます。 • ベアラー情報は、受信した PDR ID から抽出されます。 • ベアラーコンテキストは buffer-data_ind で更新されます。 • ダウンリンクデータを持つベアラーの EBI と、これらのベアラー間の最小 ARP を使用して DDN が開始されました。 • DDN を S11-GTP-EP ポッドに送信します。
6	DDN の受信後、S11-GTP-EP が次の手順を実行します。 • 新しい E-T4 トランザクションを作成します。 • DDN を MME に送信します。
7	MME は S11-GTP-EP に DDN ACK の成功を送信します。

ステップ	説明
8	ステップ 4 の後に開始されたトランザクション S-T3 が完了します。 S11-GTP-EP が SGW サービスポッドに DDN 応答の成功を送信します。
9	SGW サービスポッドが CDL を更新します。

Downlink Data Notification Failure コールフロー

ここでは、Downlink Data Notification Failure コールフローについて説明します。

図 *31 : Downlink Data Notification Failure* コールフロー

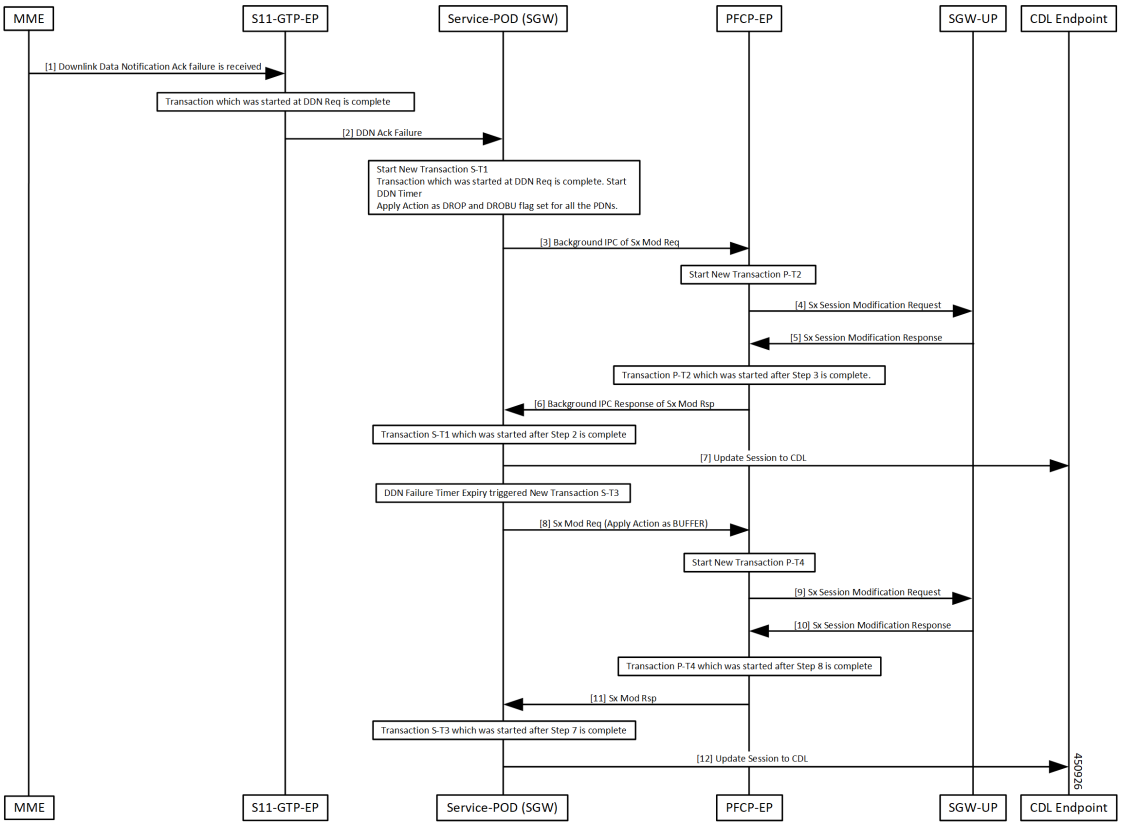


表 *54 : Downlink Data Notification Failure* コールフローの説明

ステップ	説明
1	S11-GTP-EP ポッドが DDN ACK エラーを受信します。
2	DDN 要求の送信中に開始されたトランザクションが終了します。 S11-GTP-EP が DDN ACK エラーを SGW サービスポッドに転送します。

ステップ	説明
3	SGW サービスポッドで DDN ACK エラーを受信すると、次の手順が実行されます。 • 受信した原因に基づいてページング状態を決定します。 • EGTP_CAUSE_CONTEXT_NOT_FOUND：コール削除の内部トランザクションを送信します。 • EGTP_CAUSE_UNABLE_TO_PAGE_UE • EGTP_CAUSE_UNABLE_TO_PAGE_UE_DUE_TO_SUSPENSION • EGTP_CAUSE_UE_ALREADY_REATTACHED • EGTP_CAUSE_TEMP_REJECTED_DUE_TO_HANDOVER_IN_PROGRESS • PDN が接続状態にあるか確認して、Sx Modify Request を開始します。少なくとも 1 つの PDN が CONNECTED 状態である必要があります。 • これらのページング障害の原因を処理するために、内部トランザクションを送信します。 • 現在の手順およびトランザクションを終了します。 • ページング障害を処理する新しいトランザクションでは、Sx Modify Request を送信するすべての PDN を取得します。 • ページング状態に基づいて、ページングアクションを取得し、必要なアクションに基づいて Sx Modify Request を送信します。 Sx Modification Request のバックグラウンド IPC 要求を PFCP-EP ポッドに送信します。新しいトランザクション P-T2 を作成します。
4	Sx Modification Request のバックグラウンド IPC 要求を受信後、PFCP-EP が次の手順を実行します。 • 新しい P-12 トランザクションを開始します。 • SGW-UP に送信します。
5	PFCP-EP が SGW-UP から Sx Modification Response を受信します。
6	ステップ 3 で開始されたトランザクション P-T2 が完了しました。 PFCP-EP ポッドがバックグラウンド IPC 応答を SGW サービスポッドに送信します。
7	ステップ 2 で開始されたトランザクション S-T1 が完了しました。 SGW サービスポッドが、ベアラーレベルフラグで buffer_data_ind を使用して CDL を更新しました。

ステップ	説明
8	DDN障害タイマーの期限が切れると、新しいトランザクションS-T3が開始します。 SGWサービスポッドが、適用アクションをBUFFERとして、Sx Modification Requestのバックグラウンド IPC 要求を PFCP-EP ポッドに送信します。
9	新しい P-T4 トランザクションが作成されます。 PFCP-EP ポッドが Sx Modification Request を SGW-UP に送信します。
10	SGW-UP が Sx Modification Response を PFCP-EP ポッドに送信します。
11	ステップ 8 で開始されたトランザクション P-T4 が完了しました。 PFCP-EP ポッドが Sx Modification Response を SGW サービスポッドに転送します。
12	ステップ 7 で開始されたトランザクション S-T3 が完了しました。 SGW サービスポッドが CDL を更新します。

ユーザーなし接続再試行タイマーのコールフロー

このセクションでは、ユーザーなし接続再試行タイマーのコールフローについて説明します。

図 *32 :* ユーザーなし接続再試行タイマーのコールフロー

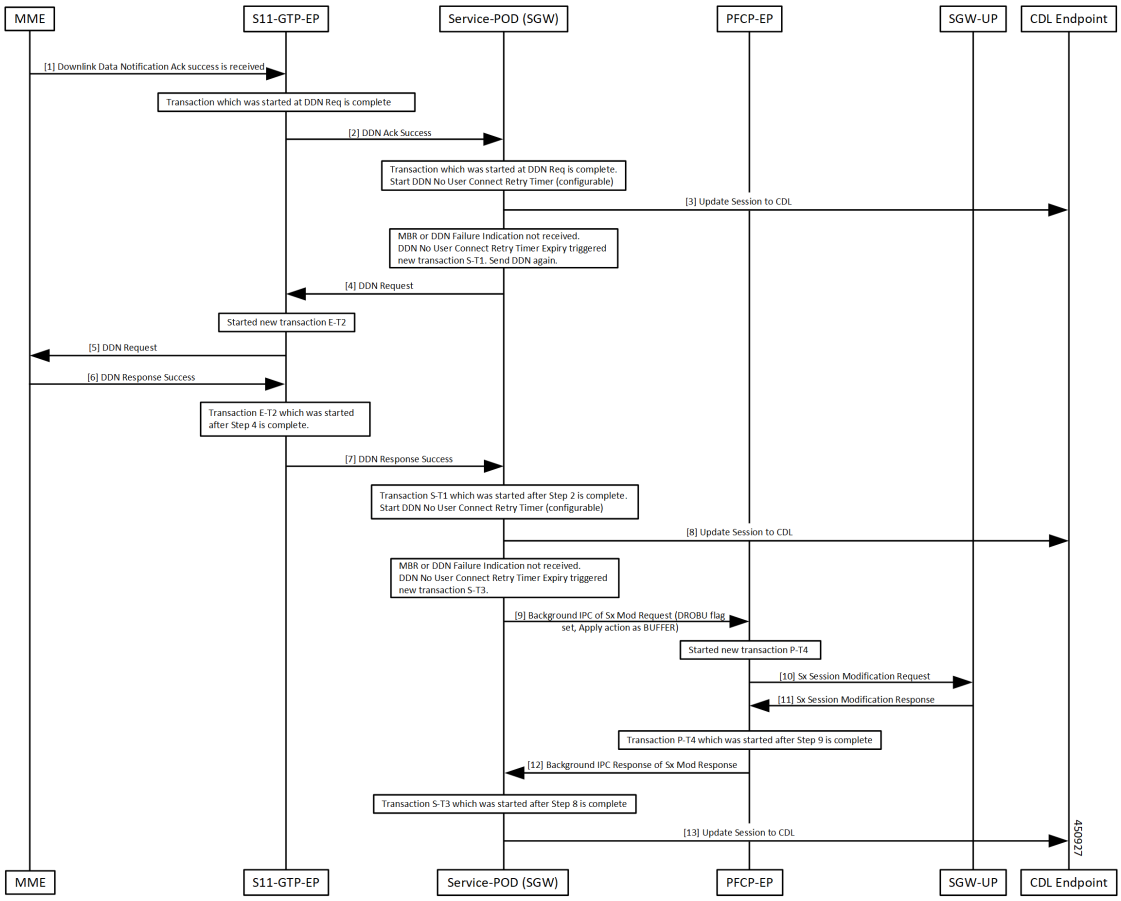


表 *55 :* ユーザーなし接続再試行タイマーのコールフローの説明

ステップ	説明
1	S11-GTP-EP ポッドで DDN ACK の成功を受信しました。
2	DDN の送信中に開始されたトランザクションはここで終了します。 S11-GTP-EP は、DDN ACK の成功を SGW サービスポッドに送信します。
3	SGW-C ポッドでユーザーなし接続再試行タイマーが開始されます。このタイマーは設定可能です。 SGW サービスポッドが CDL を更新します。

ステップ	説明
4	SGW サービスポッドは、次の場合に DDN 要求を S11-GTP-EP ポッドに送信します。 • DDN 障害通知/MBR を受信していない • ユーザーなし接続再試行タイマーが期限切れ。 新しいトランザクション S-T1 が作成されます。
5	新しい E-T2 トランザクションが作成されます。 S11-GTP-EP ポッドが DDN 要求を MME に転送します。
6	MME が S11-GTP-EP に DDN 応答を送信します。
7	ステップ 4 で開始されたトランザクション E-T2 が完了します。 S11-GTP-EP は、DDN 応答の成功を SGW サービスポッドに転送します。
8	ステップ 2 で開始された S-T1 トランザクションが完了しました。 SGW-C ポッドでユーザーなし接続再試行タイマーが開始されます。このタイマーは設定可能です。 SGW サービスポッドが CDL を更新します。
9	DDN 障害通知/MBR を受信していない場合、ユーザーなし接続再試行の有効期限がトリガーされます。 新しいトランザクション S-T3 が作成されます。 SGW サービスポッドは、Sx 変更要求のバックグラウンド IPC 要求を PFCP-EP ポッドに送信します（DROBU フラグと Apply Action を BUFFER として）。
10	新しいトランザクション P-T4 が作成されます。 PFCP-EP ポッドは SGW-U ポッドに Sx 変更要求を送信します。
11	PFCP-PE ポッドが Sx 変更応答を受信します。
12	ステップ 9 で開始されたトランザクション P-T4 が完了します。 PFCP-EP ポッドは、SGW サービスポッドにバックグラウンド IPC 応答を送信します。
13	ステップ 8 で開始されたトランザクション S-T3 が完了します。 CDL が更新されます。

機能設定

この機能の設定には、次の手順が含まれます。

DDN 障害タイマーの構成

DDN 障害タイマー（DDN Failure Timer）は sgw-profile で設定されます。

この機能を設定するには、次の構成を使用します。

```
config
   profile sgw sgw_name
      ddn failure-action-drop-timer timer_value
      ddn timeout-purge-session { true | false }
      end
```

注：

- **ddn failure-action-drop-timer** *timer_value*：DDN パケットドロップタイマーの期間を指定します。この指定されたタイムフレーム中、DDN は UE に送信されません。このタイマーは、DDN ACK 障害または DDN 障害通知の通知を受信したときに使用されます。デフォルト値は 300 秒です。

（注） タイマーを無効にするには、タイマー値をゼロに設定します。

- **ddn timeout-purge-session { true | false }**：DDN タイムアウト パージ セッションを有効または無効にするオプションを指定します。デフォルト値は false です。

設定例

次に、設定例を示します。

```
config
profile sgw sgw1
ddn failure-action-drop-timer 60
ddn timeout-purge-session false
end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw
profile sgw sgw1
locality LOC1
fqdn 209.165.201.1
ddn failure-action-drop-timer 60
ddn timeout-purge-session false
end
```

DDN No User Connect Retry Timer の設定

ここでは、DDN No User Connect Retry Timer の設定方法について説明します。

DDN No User Connect Retry Timer は、sgw-profile で設定できます。

この機能を設定するには、次の設定を使用します。

```
config
   profile sgw sgw_name
    ddn no-user-connect-retry-timer timer_value
    end
```

注：

- **ddn no-user-connect-retry-timer** *timer_value*：DDN ACK が成功し、MBR を受信しない場合に使用する DDN 再試行タイマーを指定します。デフォルト値は 60 秒です。

  このタイマーを無効にするには、値を 0 に設定します。

設定例

次に設定例を示します。

```
config
 profile sgw sgw1
  ddn no-user-connect-retry-timer 120
  end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw
profile sgw sgw1
locality LOC1
fqdn cisco.com.apn.epc.mnc456.mcc123
ddn failure-action-drop-timer 60
ddn no-user-connect-retry-timer 120
```

制御メッセージでトリガされる DDN のサポート

機能説明

この機能は、UE が IDLE モードの場合に PGW が開始した制御手順に対する UE のページングをサポートしています。

（注） この機能は CLI で制御されます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

クラウドネイティブによる PGW が開始した手順に対するダウンリンクデータ通知のコールフロー

ここでは、クラウドネイティブによる PGW が開始した手順に対するダウンリンクデータ通知のコールフローについて説明します。

図 *33*：クラウドネイティブによる *PGW* が開始した手順に対するダウンリンクデータ通知のコールフロー

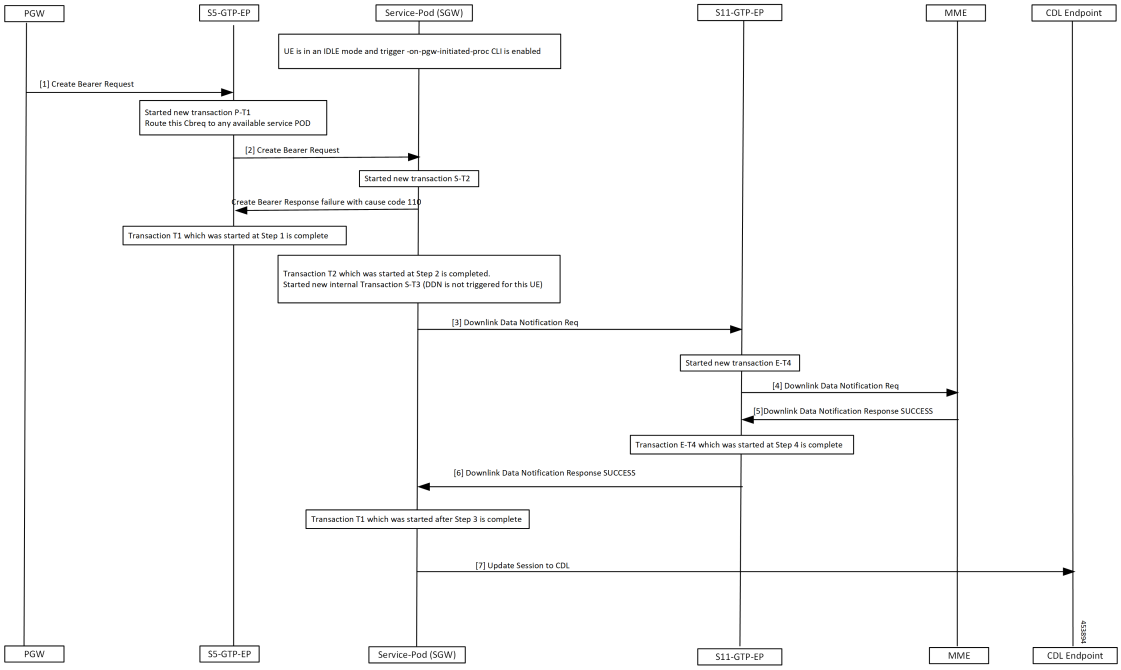


表 *56*：クラウドネイティブによる *PGW* が開始した手順（*CBR*）に対するダウンリンクデータ通知のコールフローの説明

ステップ	説明
1、2	trigger-on-pgw-initiated-proc CLI を有効にし、UE の状態が IDLE モードになります。 S5-GTP-EP は PGW から CBR を受信し、SGW サービスポッドに転送します。 SGW サービスポッドが新しい S-T2 トランザクションを開始します。 SGW サービスポッドが、原因コード 110 で S5-GTP-EP に失敗応答を送信します。
3	ステップ 2 で開始された T2 トランザクションが完了しました。 DDN がトリガーされていない UE に対して新しい S-T3 トランザクションが開始されます。 SGW サービスポッドが S11-GTP-EP への DDN 要求を開始します。

ステップ	説明
4	新しい E-T4 トランザクションが開始されます。 S11-GTP-EP が DDN 要求を MME に転送します。
5	S11-GTP-EP が MME から DDN 応答の成功を受信します。
6	ステップ 4 で開始されたトランザクション E-T4 が完了しました。 S11-GTP-EP が SGW サービスポッドに DDN 応答の成功を送信します。
7	ステップ 3 で開始されたトランザクション T1 が完了しました。 SGW サービスポッドがセッションを CDL に更新します。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
 profile sgw sgw_name
  ddn trigger-on-pgw-initiated-proc
  end
```

注：

- **ddn trigger-on-pgw-initiated-proc**：UE が IDLE モードの場合、DDN は PGW によって開始される手順のページングをトリガーします。SGW は、原因コード 110 で PGW に失敗応答を送信します。

設定例

次に、設定例を示します。

```
config
 profile sgw sgw1
  ddn trigger-on-pgw-initiated-proc
  end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw
profile sgw sgw1
locality LOC1
fqdn 209.165.201.1
ddn failure-action-drop-timer 60
ddn no-user-connect-retry-timer 120
ddn trigger-on-pgw-initiated-proc
exit
```

DDN 制御手順の無効化

DDN 制御手順機能を無効にするには、`no ddn trigger-on-pgw-initiated-proc` を使用します。

DDN Advance 機能

機能説明

この機能は次をサポートします。

- ダウンリンクデータ通知の遅延
- 高優先度ダウンリンクデータ通知
- DDN スロットリング

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

DDN 遅延コールフロー

このセクションでは、DDN 遅延コールフローについて説明します。

図 *34 : DDN* 遅延コールフロー

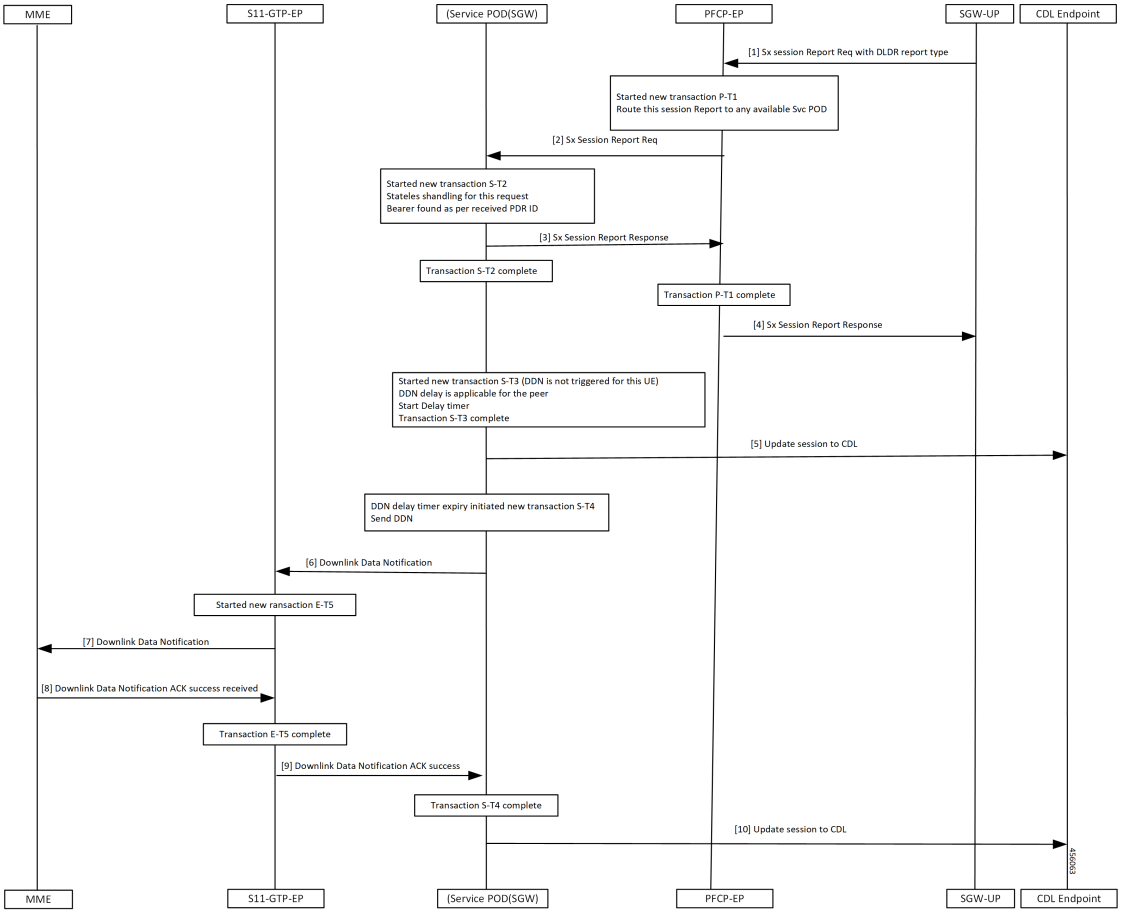


表 *57 : DDN* 遅延コールフローの説明

ステップ	説明
1	UE がアイドル状態のときに受信したダウンリンクデータ。 SGW-UP が、レポートタイプを DLDR とし、対応する PDR ID を含む Sx レポート要求を PFCP-EP に送信します。
2	新しい P-T1 トランザクションが開始されました。 PFCP-EP ポッド： • 使用可能なサービスポッドをチェックします。 • Sx セッションレポートを SGW サービスポッドに送信します。

ステップ	説明
3	新しいトランザクション S-T2 が開始されます。 SGW-CP は、CP でこの PDR-ID のベアラーが検出されると、SGW-UP に成功応答を送信します。
4	S-T2、P-T1 トランザクションが完了しました。 PFCP-EP が SGW-UP に Sx セッションレポート応答を送信します。
5	この UE に対して DDN がトリガーされていない場合、新しいトランザクション S-T3 が開始されます。 SGW サービスポッドは、ピア情報を取得して、ピアに DDN 遅延値が設定されているかどうかを確認します。 DDN 遅延が設定されている場合、DDN 遅延タイマーがトリガーされます。 R-T3 トランザクションが完了しました。 SGW サービスポッドが CDL の更新を送信します。
6、7	新しい S-T4 トランザクションが開始されます。 SGW サービスポッドが DDN を S11-GTP-EP に送信します。 新しい E-T5 トランザクションが開始されます。 S11-GTP-EP が DDN を MME に転送します。
8、9	MME が DDN ACK の成功を S11-GTP-EP に送信します。 ステップ 7 で開始されたトランザクション E-T5 が完了しました。 S11-GTP-EP が、DDN ACK の成功を SGW サービスポッドに転送します。
10	ステップ 6 で開始されたトランザクション S-T4 が完了しました。 SGW サービスポッドが、セッション情報を CDL に更新します。

高優先度 DDN コールフロー

このセクションでは、高優先度 DDN のコールフローについて説明します。

図 *35* : 高優先度 *DDN* コールフロー

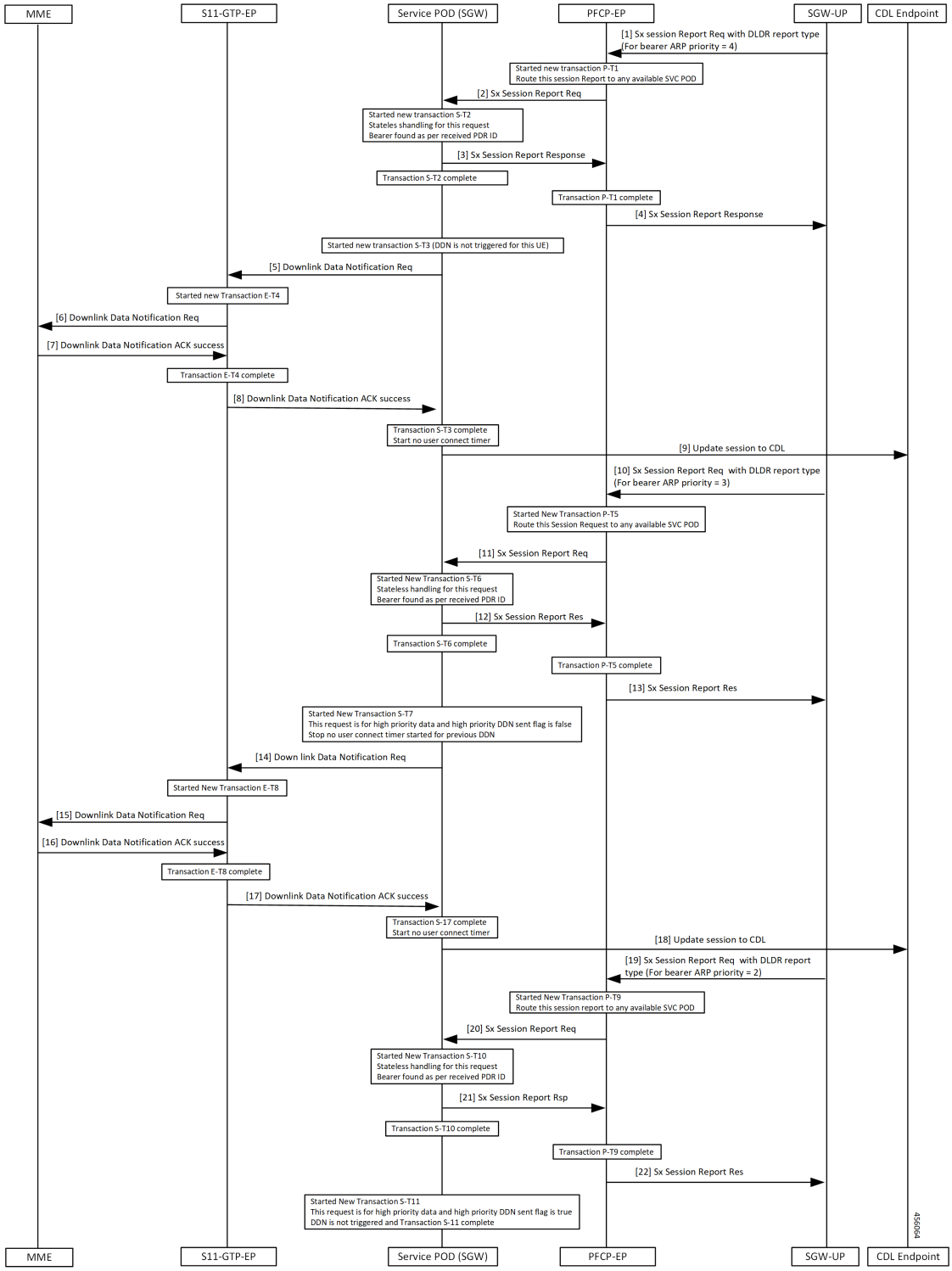


表 *58 :* 高優先度 *DDN* コールフローの説明

ステップ	説明
1	UE が IDLE 状態のときに、ベアラーが ARP 優先順位値 4 のダウンリンクデータを受信しました。 SGW-UP が、レポートタイプが DLDR で対応する PDR ID を持つように Sx レポート要求を PFCP-EP に送信します。
2	新しい P-T1 トランザクションが開始され、セッションレポートが使用可能なすべてのサービスポッドにルーティングされます。 PFCP-EP が、SGW サービスポッドに Sx セッションレポート要求を送信します。
3	新しい S-T2 トランザクションが開始され、SGW サービスポッドが Sx セッションレポート応答を PFCP-EP に送信します。
4	トランザクション P-T1 および S-T2 が完了し、PFCP-EP が Sx セッションレポート応答を SGW-UP に転送します。
5	DDN がトリガーされない新しい S-T3 トランザクションが開始されました。 SGW サービスポッドが DDN 要求を S11-GTP-EP に送信します。
6	新しい E-T4 トランザクションが開始され、S11-GTP-EP が DDN 要求を MME に転送します。
7	MME が DDN ACK の成功を S11-GTP-EP に送信します。
8	トランザクション E-T4 が完了しました。 S11-GTP-EP が、DDN ACK の成功を SGW サービスポッドに転送します。
9	トランザクション S-T3 が完了しました。 SGW サービスポッドが、No User Connect タイマーをトリガーし、CDL へのセッションを更新します。
10	SGW-UP が、ARP 優先順位値が 3 であるベアラーに対して、レポートタイプを DLDR として Sx セッションレポート要求を PFCP-EP に送信します。
11	新しい P-T5 トランザクションが開始され、セッションレポートが使用可能なすべてのサービスポッドにルーティングされます。 PFCP-EP が、SGW サービスポッドに Sx セッションレポート要求を送信します。
12	新しいトランザクション S-T6 が開始されました SGW サービスポッドが、受信した PDR ID に従ってベアラーが検出されると、Sx セッションレポート応答を PFCP-EP に送信します。

ステップ	説明
13	トランザクション S-T6 および P-T5 が完了し、PFCP-EP が Sx セッションレポート応答を SGW-UP に転送します。
14	新しいトランザクション S-T7 が開始され、データ、優先順位の高い DDN がフラグ値 False で送信されました。 上位のユーザー接続タイマーがありません。 SGW サービスポッドが DDN 要求を S11-GTP-EP に送信します。
15	新しい E-T8 トランザクションが開始され、S11-GTP-EP が MME に DDN 要求を転送します。
16	MME が DDN ACK の成功を S11-GTP-EP に送信します。
17	S11-GTP-EP は、DDN ACK の成功をこの PDR ID の SGW サービスポッドに転送します。
18	トランザクション S-17 が完了しました。SGW サービスポッドは、DDN ACK の成功を受信し、セッションを CDL に更新すると、No User Connect タイマーをトリガーします。
19	ベアラーは、ARP 優先順位値が 2 のダウンリンクデータを受信しました。 SGW-UP が、レポートタイプが DLDR で対応する PDR ID を持つように Sx レポート要求を PFCP-EP に送信します。
20	新しいトランザクション P-T9 が開始し、使用可能なすべてのサービスポッドにセッションレポートをルーティングしました。 PFCP-EP が、SGW サービスポッドに Sx セッションレポート要求を送信します。
21	新しいトランザクション S-T10 が開始され、SGW サービスポッドは、受信した PDR ID に従ってベアラーが検出されたときに、Sx セッションレポート応答を PFCP-EP に送信します。
22	トランザクション S-T10 および P-T9 が完了し、PFCP-EP は Sx セッションレポート応答を SGW-UP に転送します。 SGW サービスポッド側： • 新しいトランザクション S-T11 が開始され、データ、優先順位の高い DDN がフラグ値 True で送信されます。SGW サービスポッドが、No User Connect タイマーを停止します。 • 優先順位の高い DDN がすでに開始されている場合、SGW サービスポッドが DDN をトリガしません。トランザクション S-11 が完了しました。

DDN スロットリングコールフロー

このセクションでは、DDN スロットリングコールフローについて説明します。

図 *36 : DDN* スロットリングコールフロー

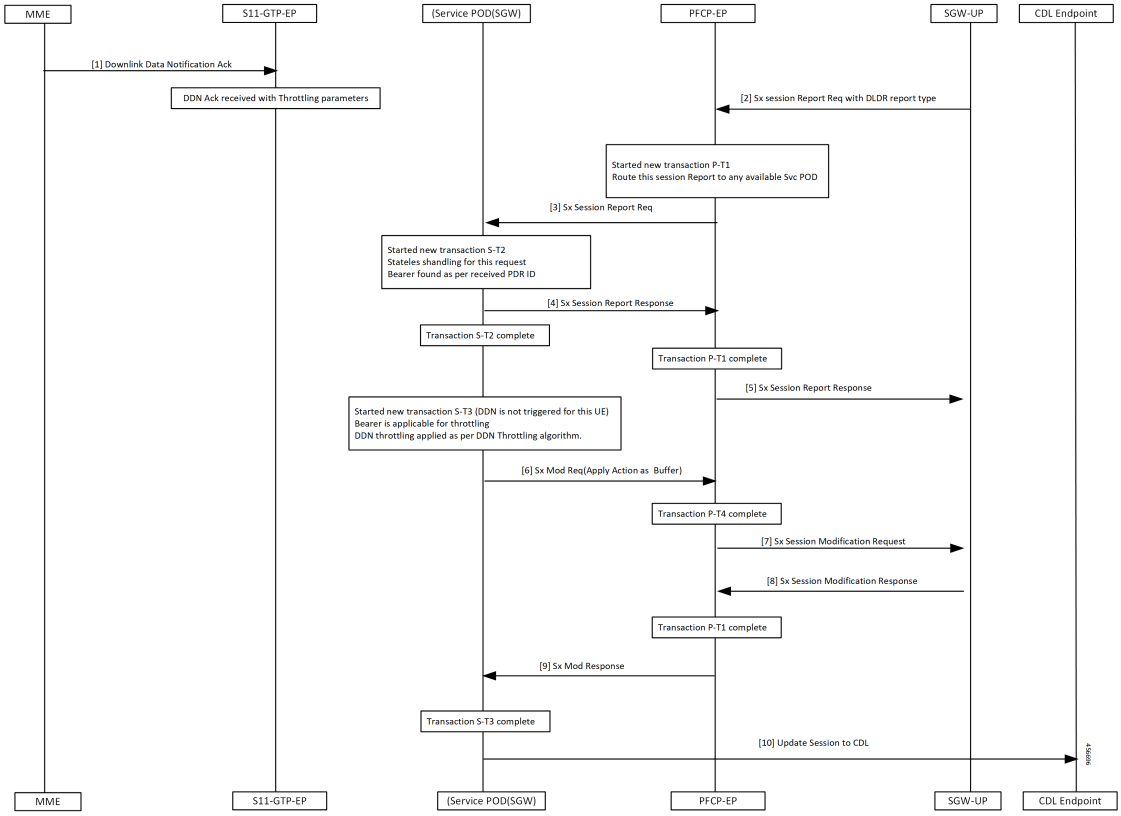


表 *59 : DDN* スロットリングコールフローの説明

ステップ	説明
1	UE がアイドル状態のときに、スロットリングパラメータとともに DDN を受信しました。 MME が DDN ACK を S11-GTP-EP に送信します。
2	SGW-UP が、レポートタイプを DLDR とし、対応する PDR ID を含む Sx レポート要求を PFCP-EP に送信します。
3	PFCP-EP が新しい P-T1 トランザクションをトリガーし、Sx レポート要求を利用可能なサービスポッドにルーティングします。 PFCP-EP が、SGW サービスポッドに Sx セッションレポート要求を送信します。

ステップ	説明
4	新しい S-T2 トランザクションを開始します。 ピア情報を取得して、このピアでDDNスロットルがアクティブかどうかを確認します。 このベアラーの優先順位が、設定されている ARP ウォーターマークより大きいかどうかを確認します。 SGW サービスポッドは、Sx セッションレポート応答を PFCP-EP に送信します。
5	ステップ 4 で開始された S-T2 トランザクションが完了しました。 ステップ 3 で開始された P-T1 トランザクションが完了しました。 この PDR ID の CP でベアラーが見つかると、PFCP-EP は SGW-UP に成功応答を送信します。
6	DDN がトリガーされていない UE に対して新しい S-T3 トランザクションが開始されます。 DDN アルゴリズムを適用して、DDN をスロットリングする必要があるかどうかを確認します。 DDN がスロットルされた場合、SGW-service ポッドは、Apply Action を BUFFER として PFCP-EP に向けて Sx 変更要求を送信します。
7、8	P-T4 トランザクションが完了しました。 PFCP-EP は、Sx セッション変更要求を SGW-UP に送信し、SGW-UP から Sx セッション変更応答を受信します。
9	ステップ 5 で開始された P-T1 トランザクションが完了しました。 PFCP-EP が Sx 変更応答を SGW サービスポッドに送信します。
10	ステップ 6 で開始された S-T3 トランザクションが完了しました。 SGW サービスポッドがセッションを CDL に更新します。

（注） cnSGW は、緊急セッションまたは WPS セッションの発信ダウンリンクデータ通知メッセージをスロットルしません。

標準準拠

Downlink Data Notification サポート機能は、次の標準に準拠しています。

- 3GPP TS 23.401「General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access」

- 3GPP TS 23.402「Architecture enhancements for non-3GPP accesses」
- 3GPP TS 29.274「3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3」
- 3GPP TS 23.214「Architecture enhancements for control and user plane separation of EPC nodes」
- 3GPP TS 29.244「Interface between the Control Plane and the User Plane nodes」
- 3GPP TS 24.008「Mobile radio interface Layer 3 specification; Core network protocols; Stage 3」

機能設定

デフォルトでは、DDN スロットリングは有効です。

（注） cnSGW-C は、MME から送信される DDN スロットリングパラメータを処理します。

この機能を設定するには、次の構成を使用します。

```
config
 profile sgw sgw_name
  ddn throttle-arp-watermark arp_value
  end
```

注：

- **ddn throttle-arp-watermark***arp_ value*：DDN スロットルの優先順位が最も低い ARP を指定します。

  スロットリングは、構成値よりも大きい ARP PL 値を持つベアラーにのみ適用されます。0 ～ 15 の範囲の整数で指定する必要があります。

  デフォルトでは、スロットリングはすべてのベアラーに適用できます。

設定例

次に、設定例を示します。

```
config
 profile sgw sgw1
  ddn throttle-arp-watermark 3
  end
```

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

DDN Advance 機能では、次の統計がサポートされています。

```
sgw_ddn_stats{app_name="smf",cluster="cn",data_center="cn",
ddn_stats_type="control_proc_triggered",instance_id="0",service_name="sgw-service"}2
```

```
sgw_ddn_stats{app_name="smf",cluster="cn",data_center="cn",ddn_stats_type="data_triggered",
 instance_id="0",service_name="sgw-service"} 18
```

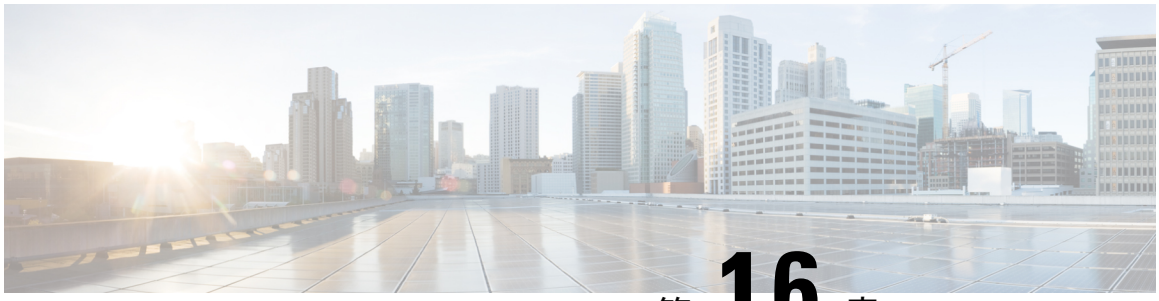
```
sgw_ddn_stats{app_name="smf",cluster="cn",data_center="cn",ddn_stats_type="delayed",
 instance_id="0",service_name="sgw-service"} 7
```

```
sgw_ddn_stats{app_name="smf",cluster="cn",data_center="cn",ddn_stats_type="high_priority_initiated",
 instance_id="0",service_name="sgw-service"} 3
```

```
sgw_ddn_stats{app_name="smf",cluster="cn",data_center="cn",ddn_stats_type="high_priority_suppressed",
 instance_id="0",service_name="sgw-service"} 1
```

```
sgw_ddn_stats{app_name="smf",cluster="cn",data_center="cn",ddn_stats_type="throttled",
 instance_id="0",service_name="sgw-service"} 6
```

- high_priority_initiated：優先順位が高いページングトリガーが原因で、DDN が開始したカウント。
- high_priority_suppressed：抑制されている DDN の優先順位が高いカウント。高い優先順位の DDN が開始したページング要求を UE がすでに処理している場合、着信の優先順位が高いページング要求を抑制します。
- throttled：DDN スロットル数。
- delayed：DDN 遅延タイマーの後に DDN が開始したカウント。
- control_proc_triggered：UE が IDLE 状態のときに制御手順から受信したページングトリガーの数。
- data_triggered：UE が IDLE 状態のときにダウンリンクデータの UPF から受信したページングトリガーの数。



第 16 章

DSCP マーキングのサポート

- 機能の概要と変更履歴 （183 ページ）
- 機能説明 （184 ページ）
- データパケットの DSCP マーキング （184 ページ）
- CP シグナリングメッセージの DSCP マーキング （187 ページ）

機能の概要と変更履歴

要約データ

表 *60* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	データパケットの DSCP マーキング：無効 - 有効にするには構成が必要 CP シグナリングメッセージの DSCP マーキング：有効 - 常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *61* : マニュアルの変更履歴

改訂の詳細	リリース
拡張および非標準（オペレータ定義）の QCI 値のサポートを検証。	2021.02.3

改訂の詳細	リリース
CP シグナリングメッセージの DSCP マーキングのサポートを追加。	2021.02.0
最初の導入。	2021.01.0

機能説明

Differentiated Services Code Point（DSCP; DiffServ コードポイント）は、ネットワークトラフィックを分類および管理する手段であり、最新のレイヤ 3 IP ネットワークで Quality of Service（QoS）を提供します。

この機能は次をサポートします。

- データパケットの DSCP マーキング
- CP シグナリングメッセージの DSCP マーキング

データパケットの DSCP マーキング

機能説明

この機能は、QCI と ARP の組み合わせによる DSCP のマーキングをサポートしています。

また、データパケットのユーザープレーン（UP）への DSCP マーキング値のプログラミングをサポートしています。

機能の仕組み

ここでは、この機能の仕組みを説明します。

DSCP マーキング IE

DSCP マーキング IE は、Sx 確立要求メッセージまたは Sx 変更要求メッセージで送信されます。これらの IE は転送アクションルール（FAR）IE の一部です。サポートされている IE とその機能は次のとおりです。

- 内部パケットマーキング（プライベート拡張 IE）：user-datagram DSCP マーキング値を UP に送信します。
- トランスポート パケット マーキング（3GPP 仕様で定義された IE）：encaps-header DSCP 値を UP に送信します。

- トランスポート パケット マーキング オプション（プライベート拡張 IE）：encaps-header マーキングの copy-inner オプションと copy-outer オプションを UP に送信します。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
   profile
      sgw-qos-profile qos_profile_name
         dscp-map
            operator-defined-qci non_standard_qos_class_id
            qci qci_value
               downlink downlink_value
                  user-datagram
                     dscp-marking dscp_marking_value
                  encaps-header
                     dscp-marking dscp_marking_value
                  encsp-header enscp_header_value
                     dscp-marking dscp_marking_value
               uplink uplink_value
                  user-datagram
                     dscp-marking dscp_marking_value
                  encaps-header
                     dscp-marking dscp_marking_value
                  encsp-header enscp_header_value
            arp-priority-level arp_priority_level_value
               uplink
                  user-datagram
                     dscp-marking dscp_marking_value
                  encaps-header
                     dscp-marking dscp_marking_value
               downlink
                  user-datagram
                     dscp-marking dscp_marking_value
                  encaps-header
                     dscp-marking dscp_marking_value
                     end
```

注：

- **sgw-qos-profile***qos_profile_name*：SGW の QoS プロファイル構成名を指定します。
- **dscp-map**：QCI から DSCP マーキングへのマッピングを構成します。
- **operator-defined-qci** *non_standard_qos_class_id*：非標準 QoS クラス識別子を指定します。128 ～ 254 の範囲の整数である必要があります。
- **qci** *qci_value*：標準の QCI 値を指定します。1 ～ 9、65、66、69、70、80、82、83 のいずれかの整数である必要があります。

- **arp-priority-level** *arp_priority_value*：ARP 優先度レベルを指定します。1 ～ 15 の範囲の整数である必要があります。
- **uplink** *uplink_value*：アップリンクの QCI 値を指定します。
- **downlink** *downlink_value*：ダウンリンク QCI 値を指定します。
- **gbr**：QCI のタイプを GBR に指定します。
- **non-gbr**：QCI のタイプを GBR 以外に指定します。
- **encaps-header**：encaps header に適用する DSCP 値を指定します。
- **user-datagram**：ユーザーデータグラムに適用する DSCP 値を指定します。
- **copy-inner**：内部 DSCP の外部値へのコピーを開始します。
- **copy-outer**：外部 DSCP の内部値へのコピーを開始します。
- **dscp-marking** *dscp_marking_value*：パケットに適用する DSCP 値を指定します。（0x で始まる 16 進文字列値。例：0x3F）
- **qci**：QCI アップリンクとダウンリンクのオプションは同じです。同様に、**operator-defined-qci** と標準 QCI のコマンドは同じです。唯一の違いは **operator-defined-qci** での *bearer-type* の選択が必須であることです。ベアラーのタイプとともに ARP を指定することもできます。

設定例

次に、設定例を示します。

```
config
  profile sgw-qos-profile q
    dscp-map qci 1 uplink encaps-header copy-inner user-datagram dscp-marking 0x1
    dscp-map qci 1 downlink user-datagram dscp-marking 0x2 encaps-header dscp-marking
0x3
    dscp-map qci 2 gbr uplink user-datagram dscp-marking 0x5 encaps-header dscp-marking
 0x6
    dscp-map operator-defined-qci 128 gbr arp-priority-level 1 uplink user-datagram
dscp-marking 0x7
    end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw-qos-profile q
  profile sgw-qos-profile q
    dscp-map qci 1 uplink encaps-header copy-inner user-datagram dscp-marking 0x1
    dscp-map qci 1 downlink user-datagram dscp-marking 0x2 encaps-header dscp-marking
0x3
    dscp-map qci 2 gbr uplink user-datagram dscp-marking 0x5 encaps-header dscp-marking
 0x6
    dscp-map operator-defined-qci 128 gbr arp-priority-level 1 uplink user-datagram
```

```
dscp-marking 0x7
    end
```

CP シグナリングメッセージの DSCP マーキング

機能説明

この機能は、次のインターフェイスでの構成に従ってパケットを制御するための DSCP 値のマーキングをサポートしています。

- GTPC：S11、S5
- PFCP：Sxa

機能設定

この機能の設定には、次の手順が含まれます。

- GTP エンドポイントの S11 インターフェイスでの DSCP の構成。詳細については、GTP エンドポイントの S11 インターフェイスでの DSCP の構成 （187 ページ） を参照してください。
- GTP エンドポイントの S5e インターフェイスでの DSCP の構成。詳細については、GTP エンドポイントの S5e インターフェイスでの DSCP の構成 （188 ページ） を参照してください。
- プロトコルエンドポイントの SxA インターフェイスでの DSCP の構成。詳細については、プロトコルエンドポイントの Sxa インターフェイスでの DSCP の構成 （189 ページ）を参照してください。

GTP エンドポイントの S11 インターフェイスでの DSCP の構成

この機能を設定するには、次の構成を使用します。

```
config
   instance instance-id instance_id
      endpoint endpoint_name
      interface interface_name
      dscp dscp_value
      end
```

注：

- **endpoint** *endpoint_name*：エンドポイント名を指定します。
- **interface** *interface_name*：エンドポイント インターフェイス名を指定します。

- **dscp** *dscp_value*：DSCP 値を指定します。0x で始まる 16 進数の文字列（0x3F など）または 10 進数の値（12 など）で指定する必要があります。10 進数値は 0 ～ 63 の範囲で指定できます。

設定例

次に、設定例を示します。

```
config
    instance instance-id 1
        endpoint gtp
        interface s11
        dscp 0x2
        end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config instance instance-id 1 endpoint
    endpoint gtp
    interface s11
    dscp 0x2
    end
```

GTP エンドポイントの S5e インターフェイスでの DSCP の構成

この機能を設定するには、次の構成を使用します。

```
config
    instance instance-id instance_id
        endpoint endpoint_name
        interface interface_name
        dscp dscp_value
        end
```

設定例

次に、設定例を示します。

```
config
    instance instance-id 1
        endpoint gtp
        interface s5e
        dscp 0x2
        end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config instance instance-id 1 endpoint
    endpoint gtp
    interface s5e
    dscp 0x2
    end
```

プロトコルエンドポイントの Sxa インターフェイスでの DSCP の構成

この機能を設定するには、次の構成を使用します。

```
config
    instance instance-id instance_id
        endpoint endpoint_name
        interface interface_name
        dscp dscp_value
        end
```

設定例

次に、設定例を示します。

```
config
    instance instance-id 1
        endpoint gtp
        interface sxa
        dscp 0x2
        end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config instance instance-id 1 endpoint
    endpoint gtp
    interface sxa
    dscp 0x2
    end
```

DSCP 構成の削除

インターフェイスまたはエンドポイントからDSCPシグナリング構成を削除すると、デフォルトのマーキングが使用されます。デフォルト値は 10 または 0xa（16 進数）です。

DSCP 構成をクリアするには、次の手順を実行します。

```
config
    instance instance-id instance_id
        endpoint endpoint_name
        interface interface_name
        no dscp
        end
```

設定例

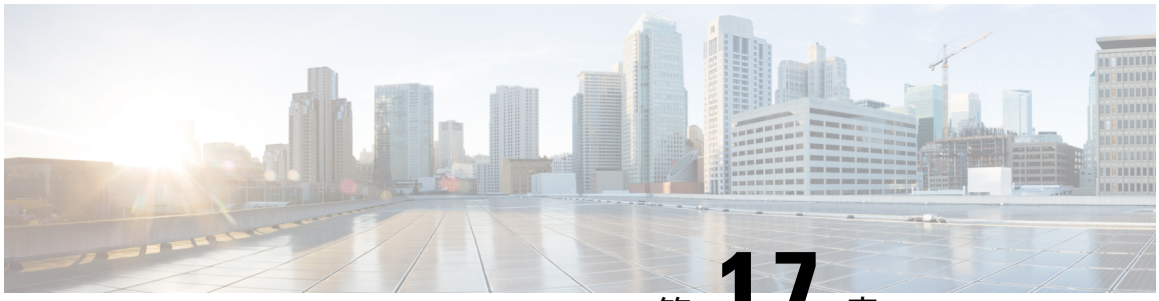
次に、DSCP 設定を削除するための設定例を示します。

```
config
    instance instance-id 1
        endpoint gtp
        interface s11
        no dscp
        end
```

設定の確認

DSCP 設定の削除を確認するには、次の手順を実行します。

```
show running-config instance instance-id 1 endpoint
    instance instance-id 1
    endpoint gtp
    interface s5e
    dscp 0x4
    exit
    interface s11
    exit
    exit
    endpoint protocol
    interface sxa
    dscp 8
    end
```



第 17 章

BGPを使用したダイナミックルーティング

- 機能の概要と変更履歴 （191 ページ）
- 機能説明 （192 ページ）
- 機能の仕組み （192 ページ）
- BGP を使用したダイナミックルーティングの設定 （201 ページ）
- モニタリングおよびトラブルシューティング （204 ページ）

機能の概要と変更履歴

要約データ

表 *62* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
デフォルト設定	無効：有効にするには設定が必要です
このリリースでの関連する変更点	N/A
関連資料	該当なし

マニュアルの変更履歴

表 *63* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

ボーダー ゲートウェイ プロトコル（BGP）では、自律システム（AS）間にループフリーのドメイン間ルーティングを作成可能です。AS は、単一の技術管理に基づくルータのまとまりです。ルータは、外部ゲートウェイプロトコルを使用して AS の外部にパケットをルーティングできます。BGP を使用したダイナミックルーティング機能を使用すると、優先順位とルートとともに IP アドレスにサービスを提供する代替ローカル アドレスを持つ BGP ルータのネクストホップ属性を構成できます。SMF BGP スピーカーポッドは、BGP を使用してポッド ルートをサービス VIP にアドバタイズすることにより、トラフィックのダイナミックルーティングを可能にします。

この機能は、次の機能をサポートしています。

- BGP を使用したダイナミックルーティングによる着信トラフィックのサービス IP アドレスのアドバタイズ。
- 発信トラフィックのルートの学習。
- BGP ポッドフェールオーバーの処理。
- プロトコル ポッド フェールオーバーの処理。
- BGP スピーカーの統計と KPI。
- BGP スピーカーをデバッグするためのログメッセージ。
- BGP スピーカーポッドの有効化または無効化。
- BGP を設定するための新しい CLI コマンド。

機能の仕組み

ここでは、ダイナミックルーティング機能の動作について説明します。

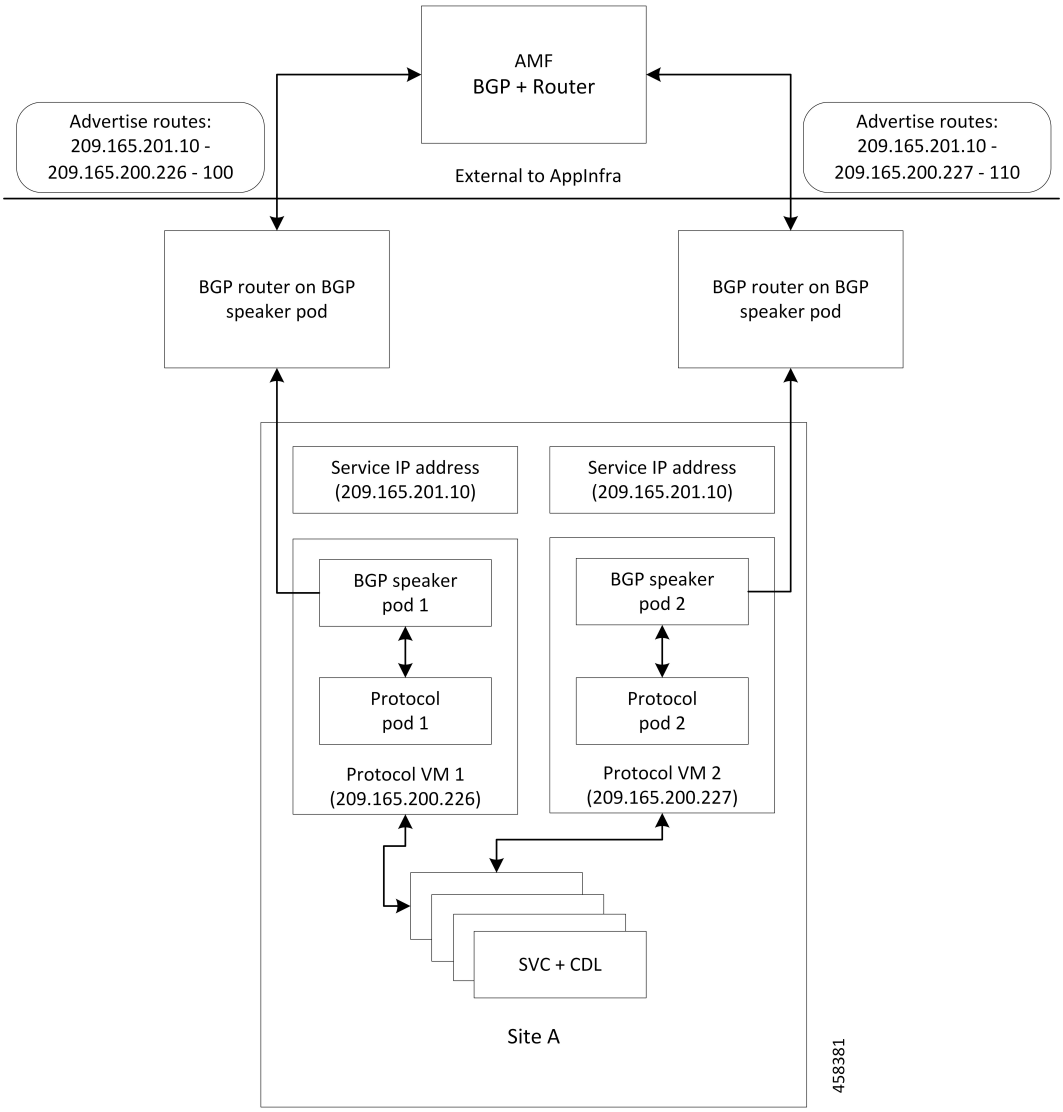
着信トラフィック

BGP は、ポート 179 でトランスポートプロトコルとして TCP を使用します。2 台の BGP ルータは相互に TCP 接続を形成し、各ルータはピアルータです。ピアルータはメッセージを交換し、接続パラメータを開いて確認します。

BGP スピーカーは、アクティブ/スタンバイモードで着信トラフィックのプロトコルポッドのルーティング情報を公開します。次の図を例として使用して、ダイナミックルーティングの機能を理解してください。pod1 と pod2 の 2 つのプロトコルポッドがあります。pod1 はアクティブで、pod2 はスタンバイモードになっています。サービス IP アドレス 209.165.201.10 は、209.165.200.226 と 209.165.200.227 の両方のノードで設定されています。pod1 はホスト 209.165.200.226 で、pod2 はホスト 209.165.200.227 で実行されています。ホスト IP アドレスはポッドサービスを公開します。BGP スピーカーは、ルート 209.165.201.10 を 209.165.200.226 と

209.165.200.227に公開します。また、ポッドの優先順位を決定するために、プリファレンス値110と100を公開します。

図 *37*：アクティブ/スタンバイトポロジでの着信トラフィックのダイナミックルーティング



高可用性を実現するため、各クラスタにはアクティブ/スタンバイトポロジを備えた2つのBGPスピーカーポッドがあります。カーネルルートの変更は、プロトコルポッドが実行されているホスト/ネットワークレベルで実行されます。

MED 値

ローカルプリファレンスはIGPネイバーでのみ使用され、MED属性はEGPネイバーでのみ使用されます。BGPでは、より低いMED値が推奨されます。

表 *64 : MED* 値

ボンディングインターフェイスがアクティブ	VIP の存在	MED 値	Local Preference
対応	対応	1210	2220
対応	非対応	1220	2210
非対応	対応	1215	2215
非対応	非対応	1225	2205

BGP スピーカーポッドのブートストラップ

次の一連の手順で、BGP スピーカーポッドをセットアップします。

1. BGP スピーカーポッドは、ポート 179 でトランスポートプロトコルとして TCP を使用し、Ops Center CLI で設定されている AS 番号を使用します。
2. Topology Manager を登録します。
3. リーダーポッドを選択します。アクティブなスピーカーポッドがデフォルトの選択肢です。
4. Ops Center CLI によって提供されるすべての BGP ピアへの接続を確立します。
5. ETCD から既存のすべてのルートを公開します。
6. CLI 設定を使用して、ルーティングのインポートポリシーを設定します。
7. 両方のスピーカーポッドで gRPC ストリームサーバーを開始します。
8. キャッシュポッドと同様に、各名前空間で 2 つの BGP スピーカーポッドを実行する必要があります。

外部ネットワーク障害

NF インスタンスの起動により、BGP スピーカー K8s ポッドは、優先順位とルートとともに、IP アドレスにサービスを提供するための代替ローカルアドレスを使用して、BGP ルータのネクストホップ属性を設定します。

地理的 HA がトリガーされると、宛先サービスの IP アドレス、パスの接続性、および優先度の値に基づいてパスが選択されます。

（注） ポッド間の透過的な移行により、サブスクライバセッションは影響を受けません。

Geo スイッチオーバー

SMF は、サービス IP アドレスを、対応するピア K8s クラスタ、コロケーションされたラック、または地理的に配置されたラックに透過的に移行することで、Geo スイッチオーバーを実現します。NF の起動中に、すべての K8s クラスタの名前空間がネクストホップ BGP ルータに登録され、そのサービス IP アドレスとローカル IP アドレスが優先順位値とルート修飾子値とともにアドバタイズされます。

各論理 NF は、名前空間の NRF または DNS に対して個別の NF インスタンス、個別の設定、および個別の LCM を公開します。

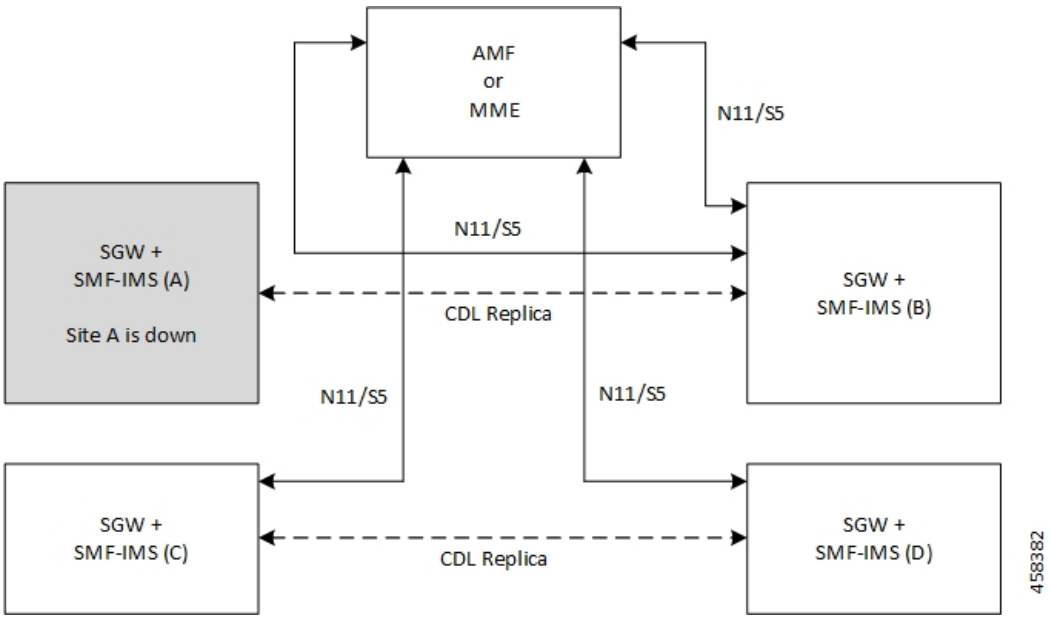
内部ネットワーク障害

マスターノードとのサーバー通信の中断、BFD の障害、または K8s ポッドのネットワークの問題が原因で、機能している K8s クラスタに内部ネットワーク障害が発生した場合、K8s の稼働状態の障害に基づく K8s の依存関係チェックにより、Geo HA がトリガーされます。

次の図に示した例では、AMF または MME が透過的に代替ラックサーバーの使用を開始しています。N11/S11/S5 および N4/Sxa サービスアドレスはサイト B のラック B に移行されます。システムはラック A のラック B からのシグナリングを継続します。ラック B では、既存のサブスクライバセッションに影響を与えることなくセッションが続行されます。

（注） UE が再接続する前にコールが終了される状態によっては、接続中のコールがほとんど失敗する場合があります。

図 *38* : 内部ネットワーク障害に対する *Geo HA*



ローカルスイッチオーバー

SMF は、同じデータセンター内に併置されたピア K8s クラスタやラックにサービス IP アドレスを透過的に移行することで、Geo スイッチオーバーを実現します。NF の起動中に、すべての K8s クラスタの名前空間がネクストホップ BGP ルータに登録され、そのサービス IP アドレスとローカル IP アドレスが優先順位値とルート修飾子値とともにアドバタイズされます。各論理 NF は、名前空間の NRF または DNS に対して個別の NF インスタンス、個別の設定、および個別の LCM を公開します。

リカバリおよびフェールバック

シームレスなフェールオーバーとフェールバックのために、UE セッションと対応するサービス IP アドレスがグループ化されます。

次のシナリオでは、UE セッションのシームレスなフェールオーバーとフェールバックのメカニズムについて説明します。

- [Normal]：UEセッションセットは、最初のラックから作成、更新、または削除され、2番目のラックに複製されます。
- [Failure]：UE セッションセットは、2 番目のラックから作成、更新、または削除されますが、可用性が低いため、最初のラックには複製されません。
- [Recovery]：すべての UE セッションデータをリカバリするために、最初のラックの CDL が 2 番目のラックの CDL と自動同期されます。リカバリ中、2 番目のラックはセッションセットからのトラフィックを処理し続けます。

コール フロー

ここでは、BGPを使用したダイナミックルーティングの主要なコールフローについて説明します。

アクティブ/スタンバイモードでの着信トラフィックのルートの公開

ここでは、アクティブ/スタンバイモードでのコントロールプレーンとデータプレーンのコールフローについて説明します。

コントロールプレーンのコールフロー

ここでは、コントロールプレーンのコールフローについて説明します。

図 *39 :* コントロールプレーンのコールフロー

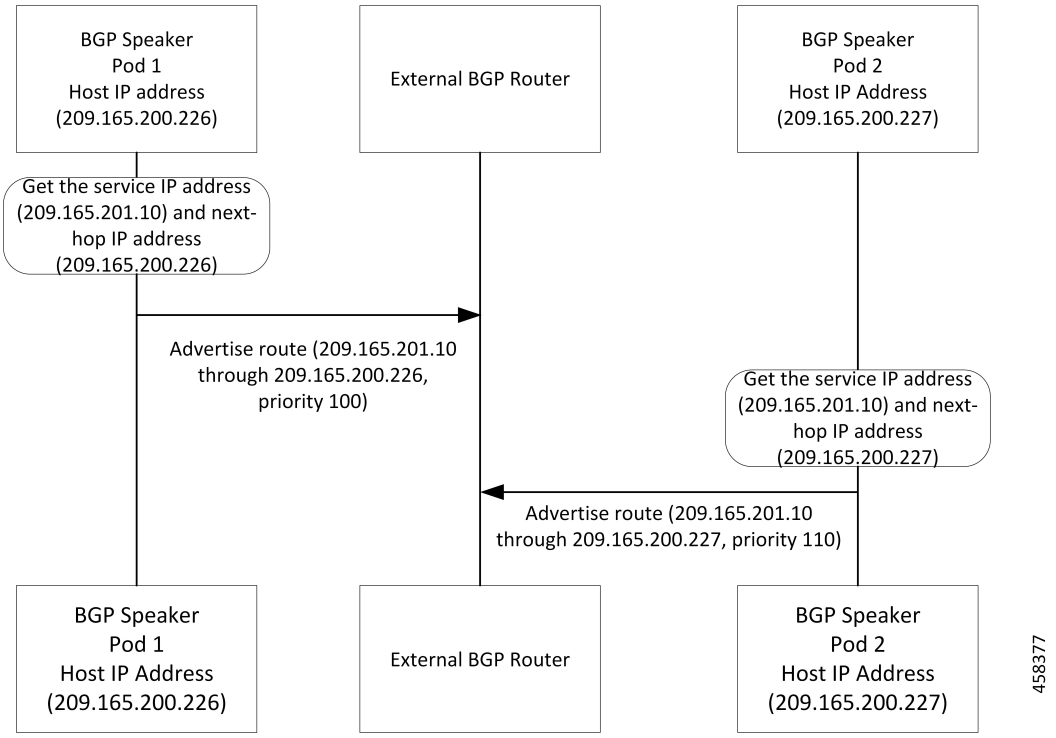


表 *65 :* コントロールプレーンのコールフローの説明

ステップ	説明
1	BGP スピーカーポッドが起動し、サービス IP アドレス、ネクストホップ IP アドレス（ホスト IP または loopbackEth）、および BGP スピーカーポッドのインスタンス ID を取得します。 ポッドサービスは、ホスト IP または設定された loopbackEth を介して公開されます。 NF インスタンス ID は、ルートの優先順位やプリファレンスを検索するために使用されます。
2	BGP スピーカーポッドは、Ops Center から vip-ip（サービス IP アドレス）を取得してルートをアドバタイズします。

データプレーンのコールフロー

ここでは、データプレーンのコールフローについて説明します。

図 *40 :* データプレーンのコールフロー

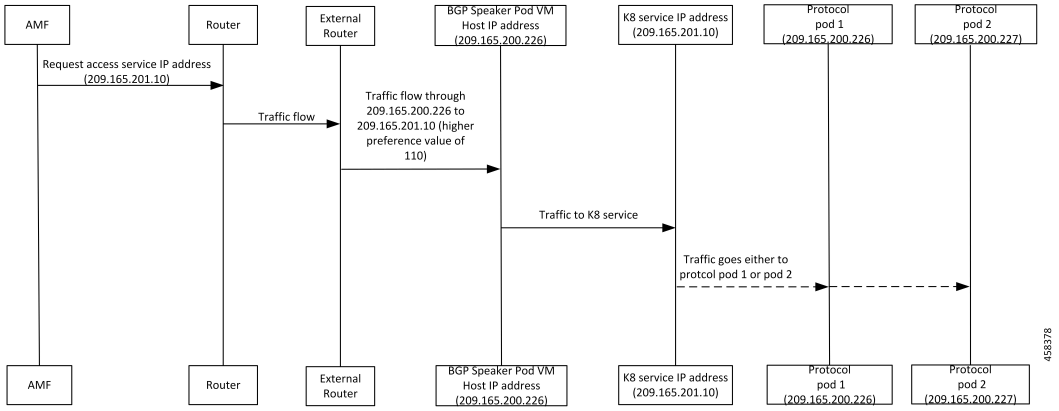


表 *66 :* データプレーンのコールフローの説明

ステップ	説明
1	サービス IP アドレスの AMF 要求。要求は、複数の外部ルータを介して、接続された最も近いルータに送信されます。次に、ルータが優先順位が最も高い BGP スピーカーポッドに要求を送信します。
2	BGP ルータは、プリファレンス値に基づいてデータプレーンフローを設定します。前述のコールフローの例では、ルータは、ホスト（209.165.200.226）を介して、プリファレンス値が高いポッド 1 にサービスリクエストをルーティングします。 ホスト（209.165.200.226）からのトラフィックは K8 サービス IP アドレス（209.165.201.10）に転送され、プロトコルポッド 1（209.165.200.226）またはポッド 2（209.165.200.227）に送信されます。

シングルプロトコルポッド障害コールフロー

ここでは、シングルプロトコルポッド障害コールフローについて説明します。

図 *41* : シングルプロトコルポッド障害コールフロー

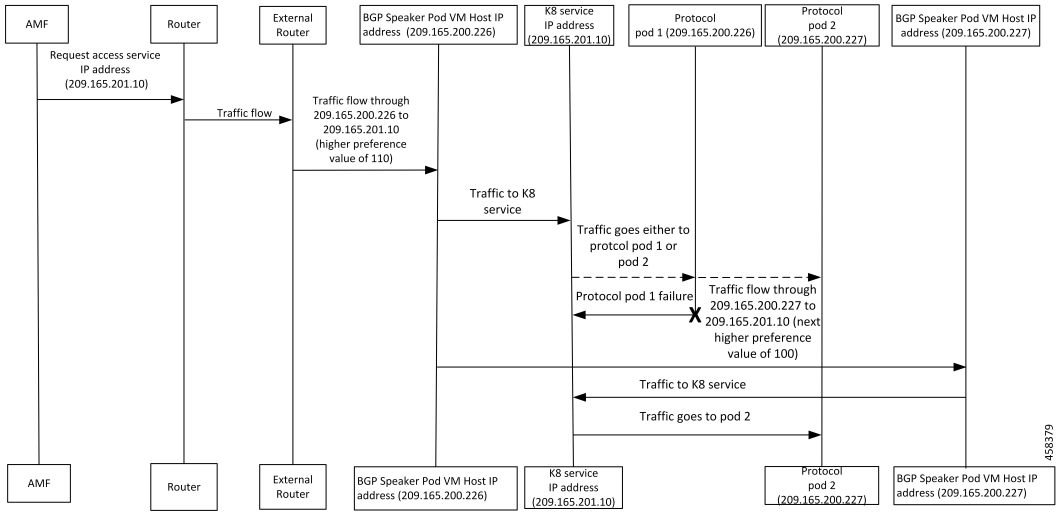


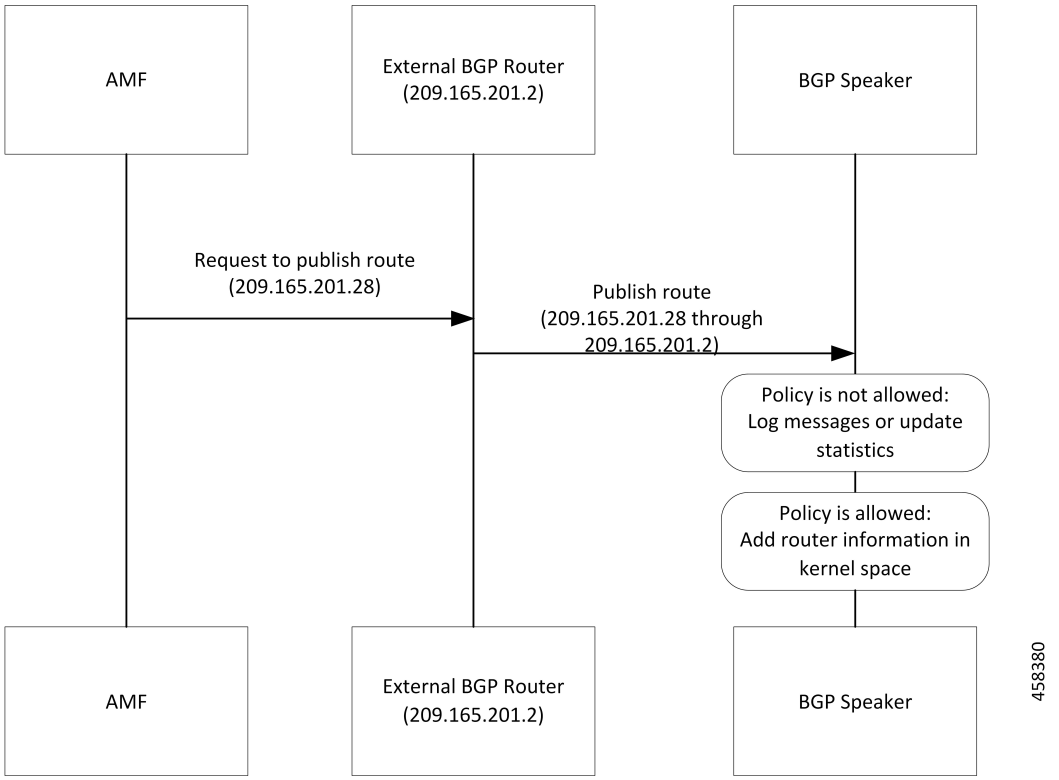
表 *67* : シングルプロトコルポッド障害コールフローの説明

ステップ	説明
1	サービス IP アドレスの AMF 要求。要求は、次に高いプリファレンス値に基づいて、複数の外部ルータを介して、接続された最も近い BGP ルータに送信されます。
2	BGP ルータは、プリファレンス値に基づいてデータプレーンフローを設定します。最も高いプリファレンス値を持つポッドが使用できない場合、要求は K8 サービスポッドを介して次に高いプリファレンス値を持つポッドにルーティングされます。 前述のコールフローの図に示した例では、IP アドレス 209.165.200.227 を持つポッド 2 が、プリファレンス値が高いため、リクエストを処理します。

発信トラフィックのルートの学習コールフロー

ここでは、発信トラフィックのルートの学習コールフローについて説明します。

図 *42 :* 発信トラフィックのルートの学習コールフロー



AMF などのシステムが外部 BGP ルートにルートをアドバタイズし、次に、外部 BGP ルータが BGP を介してそのサービスのルートをアドバタイズします。

表 *68 :* 発信トラフィックのルートの学習コールフローの説明

ステップ	説明
1	BGP スピーカーがルーティング情報を受信します。
2	BGP プロトコルを使用してルートを学習します。
3	設定ポリシーに基づいて、システムはルーティング情報を確認するか、無視します。
4	ポリシーが許可されていない場合、システムはメッセージをログに記録し、統計情報を更新します。
5	プロトコルポッドが、netlink go API を介してホスト上のカーネルスペースでルートを設定します。

BGP を使用したダイナミックルーティングの設定

ここでは、BGP を使用したダイナミックルーティングの設定方法について説明します。

AS と BGP ルータの IP アドレスの設定

BGP ルータの AS と IP アドレスを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。

  ラック間冗長性展開では、2 つの自律システム（AS）を設定する必要があります。

  - リーフとスパイン用に 1 つの AS。
  - 両方のラックの第 2 AS：Rack-1 と Rack-2。

BGP サービスリスニング IP アドレスの設定

BGP サービスリスニング IP アドレスを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      interface interface_name
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **interface** *interface_name*：インターフェイスの名前を指定します。

BGP ネイバーの設定

BGP ネイバーを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      interface interface_name
      neighbor neighbor_ip_address remote-as as_number
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。

- **interface** *interface_name*：インターフェイスの名前を指定します。
- **neighbor** *neighbor_ip_address*：ネイバー BGP ルータの IP アドレスを指定します。
- **remote-as** *as_number*：AS の ID 番号を指定します。

ボンディング インターフェイスの設定

インターフェイスに関連するボンディングインターフェイスを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      interface interface_name
      bondingInterface interface_name
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **interface** *interface_name*：インターフェイスの名前を指定します。
- **bondingInterface** *interface_name*：インターフェイスに関連するボンディング インターフェイスを指定します。ボンディング インターフェイスがアクティブである場合、BGP は、より低い MED 値を提供することで、インターフェイスサービスに高いプリファレンスを与えます。

デフォルトルートの学習の設定

システム上に特定のルートを設定し、すべてのルートをサポートする必要があるユーザーは、**learnDefaultRoute** を **true** に設定する必要があります。

（注） この設定は任意です。

デフォルトルートの学習を設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      learnDefaultRoute true/false
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **learnDefaultRoute** *true/false*：**learnDefaultRoute** パラメータを有効または無効にするオプションを指定します。true に設定すると、BGP はデフォルトルートを学習して、カーネルスペースに追加します。デフォルトは False です。

BGP ポートの設定

BGP サービスのポート番号を設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      loopbackPort port_number
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **loopbackPort** *port_number*：BGP サービスのポート番号を指定します。デフォルト値は 179 です。

ポリシーの追加

BGP スピーカーポッドは、ネイバーから多くのルート情報を学習しますが、発信トラフィックのサポートに使用されるルートはごく一部のルートです。これは、cnSGW-C が外部の AMF/PCF に情報を送信している場合、出力トラフィックの処理にのみ必要です。ルートは、BGP スピーカーでインポートポリシーを設定することでフィルタリングされ、学習したルートをプロトコルポッドに送信するために使用されます。

ポリシー追加用のサンプル CLI コードと、各パラメータに対応する説明を以下に示します。

```
$bgp policy <policy_Name> ip-prefix 209.165.200.225 subnet 16 masklength-range 21..24
as-path-set "^65100"
```

表 *69*：インポートポリシーのパラメータ

要素	説明	例	オプション
as-path-set	AS パス値	"^65100"	対応
ip-prefix	プレフィックス値	"209.165.200.225/16"	対応
masklength-range	長さの範囲	"21..24"	対応
interface	送信元 IP として設定するインターフェイス（デフォルトは VM IP）	eth0	対応
gateWay	着信ルートのゲートウェイを変更します。	209.165.201.30	対応
modifySourceIp	着信ルートの送信元IPを変更します。 デフォルト値は False です。	true	対応

要素	説明	例	オプション
isStaticRoute	カーネルルートに静的IPアドレスを追加するフラグ デフォルト値は False です。	true	対応

モニタリングおよびトラブルシューティング

ここでは、BGP を使用したダイナミックルーティング機能でサポートされている show コマンドについて説明します。

show bgp-kernel-route

BGP ルータのカーネルレベルのすべてのルートを表示するには、**show bgp-kernel-route** コマンドを使用します。

次の構成は、**show bgp-kernel-route** コマンドの出力例です。

```
kernel-route

-----bgpspeaker-pod-1 ----

 DestinationIP       SourceIP        Gateway

 209.165.200.235     209.165.200.239   209.165.200.239

-----bgpspeaker-pod-2 ----

 DestinationIP       SourceIP        Gateway

 209.165.200.235     209.165.200.229   209.165.200.244
```

show bgp-global

すべての BGP グローバル コンフィギュレーションを表示するには、**show bgp-global** コマンドを使用します。

次の構成は、**show bgp-global** コマンドの出力例です。

```
global-details

-----bgpspeaker-pod-1 ----
AS:        65000
Router-ID: 209.165.200.239
Listening Port: 179, Addresses: 209.165.200.239
AS:        65000
Router-ID: 209.165.200.232
Listening Port: 179, Addresses: 209.165.200.232

-----bgpspeaker-pod-2 ----
AS:        65000
Router-ID: 209.165.200.235
Listening Port: 179, Addresses: 209.165.200.235
```

```
AS:        65000
Router-ID: 209.165.200.246
Listening Port: 179, Addresses: 209.165.200.246
```

show bgp-neighbors

BGP ルータのすべての BGP ネイバーを表示するには、**show bgp-neighbors** コマンドを使用します。

次の構成は、**show bgp-neighbors** コマンドの出力例です。

```
neighbor-details

-----bgpspeaker-pod-2 ----
Peer              AS  Up/Down State       |#Received  Accepted
209.165.200.244 60000 00:34:20 Establ      |       10        10
Peer           AS  Up/Down State       |#Received  Accepted
209.165.200.250 60000 00:34:16 Establ      |        3         3

-----bgpspeaker-pod-1 ----
Peer              AS  Up/Down State       |#Received  Accepted
209.165.200.244 60000 00:33:53 Establ      |       10        10
Peer           AS  Up/Down State       |#Received  Accepted
209.165.200.250 60000 00:33:53 Establ      |        3         3
```

show bgp-neighbors *ip*

BGP ルータのネイバーの詳細を表示するには、**show bgp-neighbors** *ip* コマンドを使用します。

次の構成は、**show bgp-neighbors** *ip* コマンドの出力例です。

```
neighbor-details

-----bgpspeaker-pod-1 ----
BGP neighbor is 209.165.200.244, remote AS 60000
  BGP version 4, remote router ID 209.165.200.244
  BGP state = ESTABLISHED, up for 00:34:50
  BGP OutQ = 0, Flops = 0
  Hold time is 90, keepalive interval is 30 seconds
  Configured hold time is 90, keepalive interval is 30 seconds

  Neighbor capabilities:
    multiprotocol:
        ipv4-unicast:   advertised and received
    route-refresh:      advertised and received
    extended-nexthop:   advertised
        Local:  nlri: ipv4-unicast, nexthop: ipv6
    4-octet-as: advertised and received
  Message statistics:
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            70         70
    Route Refresh:          0          0
    Discarded:              0          0
    Total:                 72         73
  Route statistics:
    Advertised:             0
    Received:              10
    Accepted:              10

-----bgpspeaker-pod-2 ----
```

```
BGP neighbor is 209.165.200.244, remote AS 60000
  BGP version 4, remote router ID 209.165.200.244
  BGP state = ESTABLISHED, up for 00:35:17
  BGP OutQ = 0, Flops = 0
  Hold time is 90, keepalive interval is 30 seconds
  Configured hold time is 90, keepalive interval is 30 seconds

  Neighbor capabilities:
    multiprotocol:
        ipv4-unicast:   advertised and received
    route-refresh:      advertised and received
    extended-nexthop:   advertised
        Local:  nlri: ipv4-unicast, nexthop: ipv6
    4-octet-as: advertised and received
  Message statistics:
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            71         71
    Route Refresh:          0          0
    Discarded:              0          0
    Total:                 73         74
  Route statistics:
    Advertised:             0
    Received:              10
    Accepted:              10
```

show bgp-route-summary

BGP ルータのすべてのルートの詳細を表示するには、**show bgp-route-summary** コマンドを使用します。

次の構成は、**show bgp-route-summary** コマンドの出力例です。

```
route-details

-----bgpspeaker-pod-1 ----
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 5, Path: 5

-----bgpspeaker-pod-2 ----
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 5, Path: 5
```

show bgp-routes

BGP ルータのすべてのルートを表示するには、**show bgp-routes** コマンドを使用します。

次の構成は、**show bgp-routes** コマンドの出力例です。

```
bgp-route

-----bgpspeaker-pod-1 ----
   Network              Next Hop             AS_PATH       Age        Attrs
*> 209.165.200.235/24   209.165.200.250      60000         00:36:39   [{Origin: i} {Med:
 0}]
*> 209.165.200.227/32   209.165.200.232                    00:36:44   [{Origin: e}
{LocalPref: 220} {Med: 3220}]
*> 209.165.200.247/24   209.165.200.250      60000         00:36:39   [{Origin: i} {Med:
 0}]
*> 209.165.200.251/24   209.165.200.250      60000         00:36:39   [{Origin: i} {Med:
 0}]
```

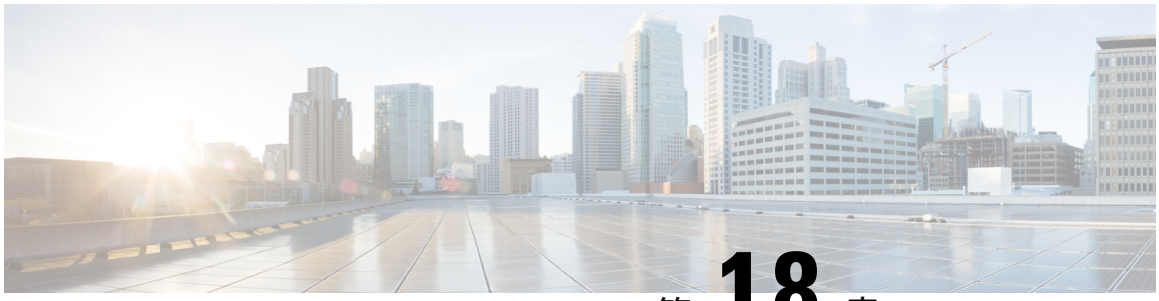
```
*> 209.165.200.252/32   209.165.200.232                    00:36:44   [{Origin: e}
{LocalPref: 220} {Med: 3220}]

-----bgpspeaker-pod-2 ----
   Network              Next Hop             AS_PATH        Age        Attrs
*> 209.165.200.235/24   209.165.200.250      60000          00:37:02   [{Origin: i} {Med:
 0}]
*> 209.165.200.227/32   209.165.200.246                     00:37:11   [{Origin: e}
{LocalPref: 220} {Med: 3220}]
*> 209.165.200.228/24   209.165.200.234      60000          00:37:02   [{Origin: i} {Med:
 0}]
*> 209.165.200.229/24   209.165.200.234      60000          00:37:02   [{Origin: i} {Med:
 0}]
*> 209.165.200.230/32   209.165.200.246                     00:37:11   [{Origin: e}
{LocalPref: 220} {Med: 3220}]
```

KPI

この機能では、次の KPI がサポートされています。

表 *70 : BGP* を使用したダイナミックルーティングの統計情報

KPI 名	タイプ	説明/式	ラベル
bgp_outgoing_route request_total	カウンタ	受信ルートの総数。	local_pref、med、next_hope、service_IP
bgp_outgoing_failedroute request_total	カウンタ	障害の発生した受信ルートの総数。	local_pref、med、next_hope、service_IP
bgp_incoming_route request_total	カウンタ	着信ルートの総数。	interface、next_hope、service_IP
bgp_incoming_failedroute request_total	カウンタ	障害の発生した着信ルートの総数。	interface、next_hope、service_IP
bgp_peers_total	カウンタ	追加されたピアの合計数。	peer_ip、as_path
bgp_failed_peerstotal	カウンタ	障害が発生したピアの総数。	peer_ip、as_path、error



第 18 章

緊急コール サポート

- 機能の概要と変更履歴 （209 ページ）
- 機能説明 （210 ページ）
- 機能の仕組み （210 ページ）
- OAM サポート （212 ページ）

機能の概要と変更履歴

要約データ

表 *71* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *72* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

緊急サービスとは、緊急サービスをサポートするようにネットワークが設定されている場合に、サービスを提供しているネットワークによって提供される機能を指します。緊急サービスは、IMS 緊急セッションをサポートするために提供されています。

MME 緊急設定データには、PDN GW の取得に使用される緊急 APN が含まれ、緊急 APN 用に静的に設定された PDN GW を含めることもできます。

cnSGW-C では、次の場合にコールが緊急だと見なされます。

- Create Session Request に IMEI のみ含まれている。
- 通知フラグが認証されていない IMSI を示し、Create Session Request に有効な IMSI と IMEI がある。

（注） 緊急セッションの設定では、cnSGW-C は MME によって送信された追加の PDN 要求（Create Session Request）を拒否します。

制限事項

2021.02.0 以降のリリースでは、この機能には次の制限があります。

- 15 桁または 16 桁の IMEI は、次の手順（show subscriber、clear subscriber、および monitor subscriber）でのみサポートされています。

機能の仕組み

ここでは、この機能の仕組みを説明します。

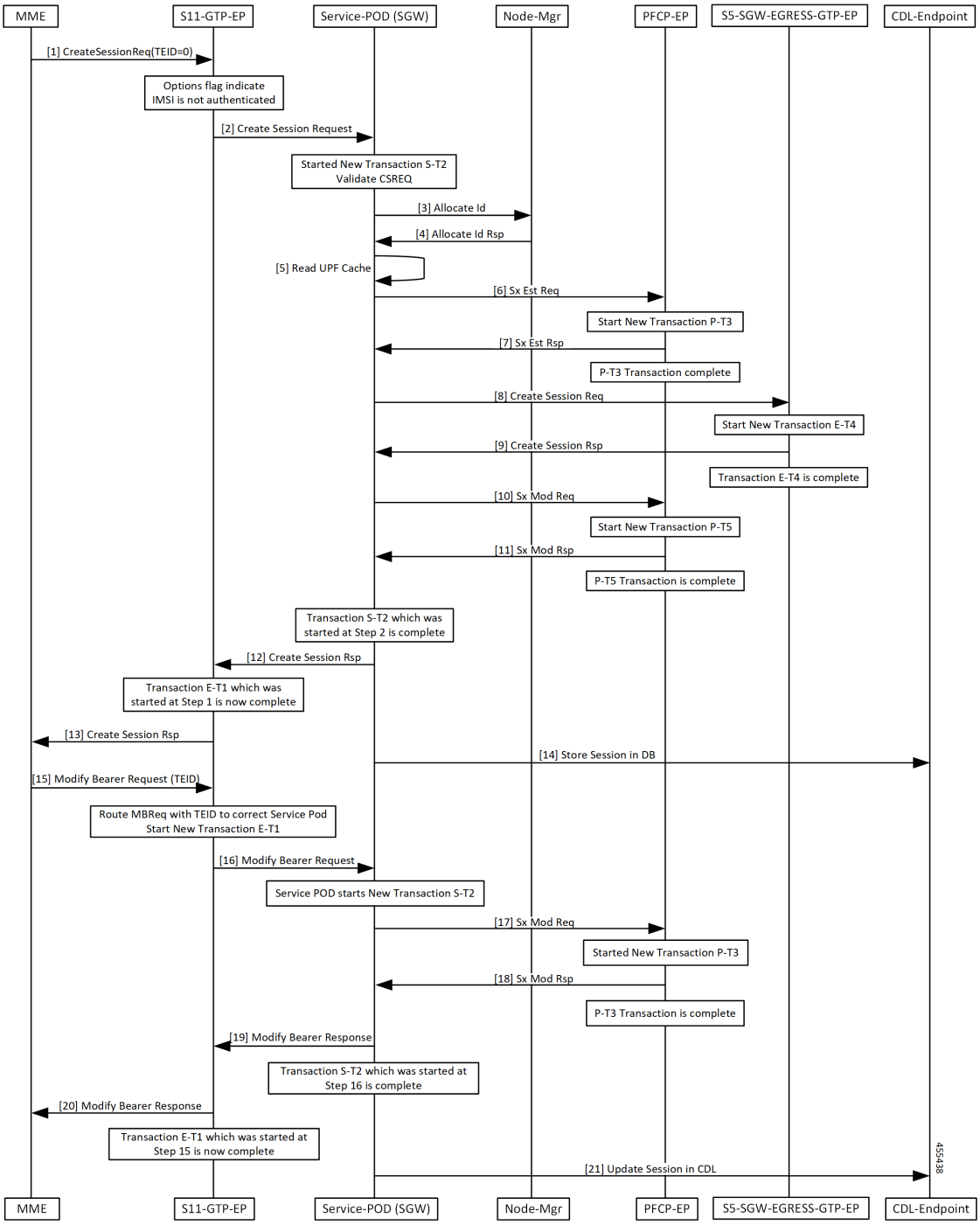
コール フロー

ここでは、この機能の主要なコールフローについて説明します。

緊急セッション作成コールフロー

このセクションでは、緊急セッション作成（IMEI、非認証 IMSI）コールフローについて説明します。

図 *43* : 緊急セッション作成コールフロー



緊急コールの場合、cnSGW-C は、CSR で未認証 IMSI または IMEI のみの初期接続要求を受信します。cnSGW-C は、ノードマネージャから TEID と SEID を割り当て、ローカル SEID を含む Sx 確立要求を UP に送信してセッションを確立します。

cnSGW-C が UP SEID と S5 のローカル GTPU TEID と S1 GTPU FTEID を含む Sx 確立応答を UP から受信すると、cnSGW-C は EGTP EP を使用して PGW にセッション作成要求を送信します。セッション作成応答に対する応答を PGW から受信した後、cnSGW-C は Sx 変更要求を送信して、PGW-U と SGW-U の間の S5-GTPU トンネルを接続します。Sx 変更応答を正常に受信すると、cnSGW-C はセッション作成応答を MME に送信し、セッションが CDL 内に作成されます。

初期接続手順では、cnSGW-C は、eNodeB と SGW-U の間の S1 GTPU トンネルを接続するベアラー変更要求の処理をサポートします。cnSGW-C が MBR を受信すると、Sx 変更要求を送信して、eNodeB と SGW-U の間の S1 GTPU トンネルを接続します。Sx 変更応答を受信した後、cnSGW-C は MME にベアラー変更応答を送信します。セッションは、トランザクションの終了時に CDL で更新されます。

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計サポート

緊急カウンタ

```
sgw_ue_stats{app_name="SMF",cluster="Local",data_center="DC",gr_instance_id="1",

instance_id="0",rat_type="EUTRAN",service_name="sgw-service",status="emergency_release"}
 9
```

```
sgw_ue_stats{app_name="SMF",cluster="Local",data_center="DC",gr_instance_id="1",

instance_id="0",rat_type="EUTRAN",service_name="sgw-service",status="emergency_setup"}
 9
```

緊急統計

```
db_records_total{app_name="datastore-ep",cluster="session",data_center="test",db=

"session",instance_id="3232257055",service_name="datastore-ep",session_type="SGW:
 emergency_call:true",sliceName="1",systemId=""} 1
```

```
db_records_total{app_name="datastore-ep",cluster="session",data_center="test",db=

"session",instance_id="3232257055",service_name="datastore-ep",session_type="SGW:
 rat_type:EUTRAN",sliceName="1",systemId=""} 3
```

```
db_records_total{app_name="datastore-ep",cluster="session",data_center="test",db=
```

```
"session",instance_id="3232257055",service_name="datastore-ep",session_type="SGW:
 state:active",sliceName="1",systemId=""} 3
```

```
db_records_total{app_name="datastore-ep",cluster="session",data_center="test",db=

"session",instance_id="3232257055",service_name="datastore-ep",session_type="total",
 sliceName="1",systemId="1"} 3
```

第 19 章

セッションの適切な終了のための拡張PFCP関連付け解除手順

- PFCP 関連付けリリースを介した UPF から cnSGWc へのコール切断通知 （216 ページ）
- UPF が開始した PFCP セッションのリリース （217 ページ）
- UPF によって開始された拡張 PFCP 関連付け解除 （218 ページ）
- PFCP セッションリリース手順を開始するための EPFAR の有効化 （219 ページ）
- バルク統計 （219 ページ）

PFCP 関連付けリリースを介した UPF から cnSGWc へのコール切断通知

表 *73 :* 機能の履歴

機能名	リリース	説明
UPF によって開始された PFCP 関連付け解除に関する通知	2024.03.0	この機能により、cnSGWc は UPF によって開始された PFCP 関連付け解除セッションの手順に関する通知を受信できます。この通知は、UPF と cnSGWc でセッションと関連付けを同時にクリアすることを示しています。 cnSGWc が通知を受信しない場合、UPF が cnSGWc から次のセッション変更要求を受信するまで、コールは接続されたままになります。この結果、サブスクライバの使用状況レポートが失われます。ここで、拡張 PFCP 関連付け解除機能（EPFAR）を使用すると、シグナリング効率が高まり、cnSGWc による使用状況レポートの効果的な処理が実現します。 **デフォルト設定**：無効：有効にするには設定が必要です

UPF がエラーまたは部分的な障害によりコールをクリアすることを決定すると、UPF はコールのクリアについて cnSGWc に通知せずに、コールをローカルでクリアします。UPF が cnSGWc から次のセッション変更要求を受信するまで、コールは接続されたままになります。

コールクリア中の使用状況レポートの損失を防ぎ、シグナリング効率を向上させるために、UPF に拡張 PFCP 関連付け解除（EPFAR）機能を適用し、PFCP 関連付け解除を開始して、cnSGWc と UPF 間のセッションを同時に正常にクリアします。この機能は、3GPP TS 29.244 のリリース 16.9.0、セクション 5.18.1 およびセクション 5.18.2 に準拠しています。

EPFAR ネゴシエーション

EPFAR ネゴシエーション機能は、CP 機能と UP 機能間の通信を処理します。CP 機能と UP 機能の両方が EPFAR 機能をサポートしている場合、関連付けリリースアクションが実行されます。UPF が SMF との関連付けを解除する必要がある場合は、構成コマンドを使用して EPFAR 機能を有効にすることもできます。

SGW は、UPF への PFCP 関連付け設定応答で CP Function Features IE の EPFAR ビットを設定することで、EPFAR 機能のサポートを示します。

ローリングアップグレードの場合、シャーシがアクティブになるたびに、EPFAR 機能が再度ネゴシエートできるように、UPF または cnSGWc は UP または CP 機能による PFCP 関連付けの更新を送信します。

UPF によって開始されたリリースは、次の手順を使用してトリガーされます。

1. UPF が開始した PFCP セッションのリリース
2. UPF によって開始された拡張 PFCP 関連付け解除

UPF が開始した PFCP セッションのリリース

1. UPF は、関連付けセットアップ手順中にネゴシエートされた場合にのみ、ピアノードの機能を有効にします。UPF は、エラーまたは部分的な障害により PFCP セッションを削除する必要がある場合、影響を受けるセッションの PFCP セッションレポート要求を開始します。
2. cnSGWc は、レポートタイプとユーザーレポートトリガーとともに UPF から PFCP セッションレポート要求を受信します。
   - PFCP セッションの使用状況がゼロ以外のレポートがある場合、レポートタイプは USAR（使用状況レポート）に設定され、送信する使用状況レポートがない場合は UISR（UP 開始セッション要求）に設定されます。

     レポートタイプ IE のオクテット 6 の 5 番目のビットは、PFCP セッションレポート要求の UISR を示すために使用される独自の IE ビットです。
   - ユーザーレポートトリガーは、ゼロ以外の使用状況レポートの TEBUR（UP 機能による終了レポート）に設定されます。
3. UPF は、PFCP セッションレポート要求メッセージでピアノードに原因 IE を送信します。IE は、セッション削除の切断理由を伝達するために次の値を使用します。
   - 201：サブスクライバによるクリア
   - 202：UP によって開始された関連付け解除
   - 203：復元失敗
   - 205：PFCP 原因自己保護での終了

Gz インターフェイスは、次に示す cnSGWc からの原因を伝達します。

UPF からの原因	Gz CauseForRecClosing
サブスクライバによるクリア（201）	通常リリース
UPによって開始された関連付け解除（202）	通常リリース
復元失敗（203）	異常なリリース
IP 送信元違反（204）	通常リリース
PFCP 原因自己保護での終了（205）	通常リリース

4. UPF が PSDBU（PFCP セッションが UP 機能によって削除されました）フラグを 1 に設定して、PFCP セッションの削除を示します。
5. cnSGWc が、リリース手順の一部として次の使用状況レポートを処理します。
   - PSDBU の TEBUR
   - UISR および PSDBU のセッションレポート
6. 使用状況レポートは、セッションリリース時に CGF にまとめて報告されます。
7. cnSGWc が PFCP セッションレポート応答を UPF に送信します。セッションが UPF でローカルに削除されると、cnSGWc は切断手順を続行します。

UPF によって開始された拡張 PFCP 関連付け解除

1. cnSGWc と UPF の両方が EPFAR 機能をサポートしている場合、UPF は関連付け解除の影響を受けるすべてのセッションに対して PFCP セッションレポート要求を開始し、それらのセッションを削除します。
2. cnSGWc は、PFCP の関連付け解除時に UPF から PARPS（PFCP 関連付け解除準備開始）フラグセットとともに PFCP 関連付け更新要求を受信し、cnSGWc は UPF の選択を停止します。
3. cnSGWc は、UPF から次のフラグセットとともに使用状況レポートを受信します。
   - ゼロ以外の最終使用状況レポートが存在するすべてのセッションに対する PSDBU を含む TEBUR。
   - 使用状況レポートがないすべてのセッションの UISR および PSDBU フラグを含むセッションレポート。
4. cnSGWc は、URSS フラグが 1 に設定された状態で、UPF から PFCP 関連付け更新要求を受信します。これは、影響を受ける PFCP セッションのゼロ以外のすべての使用状況レポー

トが送信されることを示します。cnSGWc は、この UPF の残りのすべてのセッションを削除します。

5. UPF のすべてのセッションがクリアされると、cnSGWc は UPF に対して PFCP 関連付け解除手順をトリガーして、PFCP 関連付けをクリアします。

PFCP セッションリリース手順を開始するための EPFAR の有効化

CLI の手順を使用して、EPFAR 機能を有効化できます。

手順

ステップ 1 コンフィギュレーション モードで **profile converged-core cc supported-features** コマンドを入力します。

例：

```
[sgw] sgw(config)# profile converged-core cc supported-features [ epfar ]
```

（注）
epfar：Enhanced PFCP Association Release のサポートを有効にします。

ステップ 2 **show running-config profile converged-core** show コマンドを使用して、EPFAR 機能が有効になっていることを確認します。

例：

```
[sgw] sgw# show running-config profile converged-core
Mon May  6  03:09:36.813 UTC+00:00
profile converged-core cc
 supported-features [ epfar ]
exit
[sgw] sgw#
```

バルク統計

次の統計が新しいラベルで更新されます。

- sgw_pdn_disconnect_stats、sgw_ue_disconnect_stats

クエリの例 1：

```
sgw_pdn_disconnect_stats{app_name="smf",
cluster="Local",data_center="DC",gr_instance_id="1",
instance_id="0",pdn_type="ipv4",rat_type="EUTRAN",
reason="userplane_initiated_clear_subscriber",
service_name="sgw-service"} 1
```

クエリの例 2：

```
sgw_ue_disconnect_stats{app_name="smf",cluster="Local",
data_center="DC",gr_instance_id="1",instance_id="0",
peer_ip="",reason="userplane_initiated_clear_subscriber",
service_name="sgw-service"} 1
```

ラベル：

- userplane_initiated_clear_subscriber
- userplane_initiated_association_release
- userplane_initiated_recovery_failure
- sx_urss_association_release
- userplane_initiated_self_protection_termination

- sgw_service_stats

クエリの例 1：

```
sgw_service_stats{app_name="smf",cluster="Local",
data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_egress",
reject_cause="",service_name="sgw-service",
sgw_procedure_type="upf_initiated_clear_subscriber",
status="attempted",sub_fail_reason="",svc_to_svc="False"} 1
```

クエリの例 2：

```
sgw_service_stats{app_name="smf",
cluster="Local",data_center="DC",
fail_reason="",gr_instance_id="1",
instance_id="0",interface=
"interface_sgw_egress",reject_cause="",
service_name="sgw-service",sgw_procedure_type=
"upf_initiated_clear_subscriber",status="success",
sub_fail_reason="",svc_to_svc="False"} 1
```

クエリの例 3：

```
sgw_service_stats{app_name="smf",
cluster="Local",data_center="DC",fail_reason="",
gr_instance_id="1",instance_id="0",interface=
"interface_sgw_ingress",reject_cause="",
service_name="sgw-service",sgw_procedure_type=
"upf_initiated_clear_subscriber",status="attempted",
sub_fail_reason="",svc_to_svc="False"} 1
```

クエリの例 4：

```
sgw_service_stats{app_name="smf",
cluster="Local",data_center="DC",fail_reason="",
gr_instance_id="1",instance_id="0",interface=
"interface_sgw_ingress",reject_cause="",
service_name="sgw-service",sgw_procedure_type=
"upf_initiated_clear_subscriber",status="success",
sub_fail_reason="",svc_to_svc="False"} 1
```

ラベル：

- upf_initiated_clear_subscriber

- upf_initiated_association_release
- upf_recovery_failure
- sx_urss_association_release_initiated_deletion
- userplane_initiated_self_protection_termination

- sgw_sx_session_report_stats

  クエリの例 1：

  ```
  sgw_sx_session_report_stats{app_name="smf",
  cluster="Local",data_center="DC",gr_instance_id="1",
  instance_id="0",reason="psdbu",service_name="sgw-service",
  status="success",sx_session_report_type="USAR"} 1
  ```

  クエリの例 2：

  ```
  sgw_sx_session_report_stats{app_name="smf",
  cluster="Local",data_center="DC",gr_instance_id="1",
  instance_id="0",reason="psdbu",service_name="sgw-service",
  status="success"sx_session_report_type="UISR"} 1
  ```

  ラベル：

  - USAR
  - UISR

- sgw_sx_usage_report_stats

  クエリの例：

  ```
  sgw_sx_usage_report_stats{app_name="smf",
  cluster="Local",data_center="DC",gr_instance_id="1",
  instance_id="0",service_name="sgw-service",status="success",
  sx_usage_report_trigger_type="TEBUR"} 1
  ```

  Label:

  - TEBUR

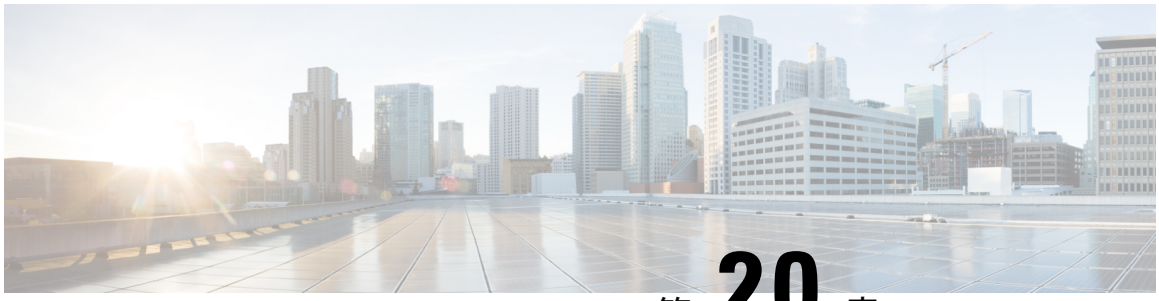
- nodemgr_up_pathfail_reasons

  クエリの例：

  ```
  nodemgr_up_pathfail_reasons {app_name="SMF",
   cluster="SMF",data_center="DC",gr_instance_id="1",
  instance_id="0",service_name="nodemgr",
  up_pathfail_reason="up_urss_reason_association_release"} 1
  ```

  Label:

  - up_urss_reason_association_release



第 20 章

拡張および非標準 QCI 値のサポートと検証

- 機能の概要と変更履歴 （223 ページ）
- 機能説明 （223 ページ）

機能の概要と変更履歴

要約データ

表 *74* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	

マニュアルの変更履歴

表 *75* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.3

機能説明

この機能は次をサポートします。

- CSR | CBR | UBR | MBC の一部としての拡張および非標準 QCI 値コールフロー

- DSCP マーキングの拡張および非標準 QCI 値
- VoLTE マーキングの拡張および非標準 QCI 値

拡張および非標準 QCI 値の検証

次の拡張 QoS クラス識別子（QCI）と非標準 QCI（オペレータ定義）の値が検証されます。

- **CSR | CBR| UBR| MBC コールフロー**： 次の QCI 値に対して検証が行われます。

  拡張 QCI： 65、66、69、70、80、82、83

  非標準（オペレータ定義）QCI： 128 ～ 154

（注） CSR | CBR | UBR | MBC コールフローの一環として、標準/拡張/オペレータ定義の QCI 範囲以外の値も受け入れられます。ただし、これらの値は、DSCP マーキングまたは VoLTE マーキングの CLI の一部としてはサポートされていません。

- **DSCP マーキング**：次の QCI 値に対して検証が行われます。

  拡張 QCI：65、66、69、70、80、82、83

  非標準（オペレータ定義）QCI：128 ～ 154

  DSCP マーキングの詳細については、『*UCC* サービング ゲートウェイ コントロール プレーンの機能 - 構成および管理ガイド』の「*DSCP* マーキングのサポート」の章を参照してください。

VoLTE マーキングの拡張および非標準 QCI 値のサポートと検証

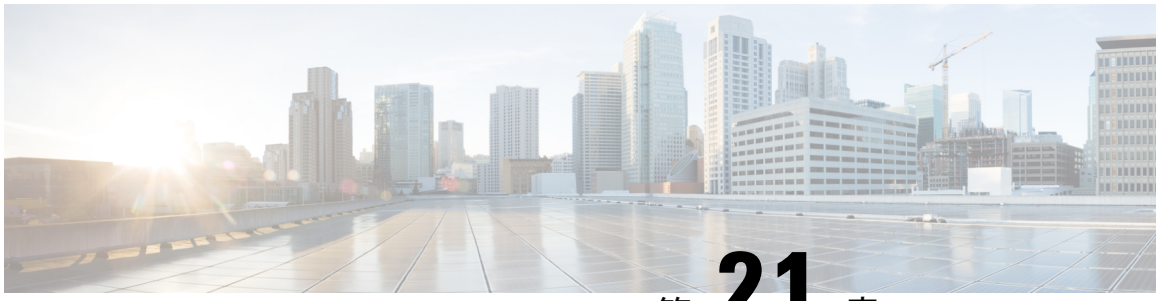
2021.02.2 より前のリリースでは、VoLTE マーキングでは標準 QCI 値 1 ー 9 のみがサポートされていました。

2021.02.2 以降のリリースでは、次の拡張および非標準 QCI 値のサポートが追加および検証されます。

拡張 QCI： 65、66、69、70、80、82、83

非標準（オペレータ定義）QCI： 128 ～ 154

VoLTE マーキングの詳細については、『UCC サービング ゲートウェイ コントロール プレーンの機能 - 構成および管理ガイド』の「VoLTE コールの優先順位付け」の章を参照してください。



第 21 章

eMPS および WPS のサポート

- 機能の概要と変更履歴 （225 ページ）
- 機能説明 （226 ページ）
- eMPS/WPS サポート （226 ページ）
- 機能設定 （233 ページ）
- OAM のサポート （238 ページ）

機能の概要と変更履歴

要約データ

表 *76* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	無効：有効にするには設定が必要です
関連資料	該当なし

マニュアルの変更履歴

表 *77* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

この機能は次をサポートします。

- Enhanced Multimedia Priority Service（eMPS）または Wireless Priority Service（WPS）
- eMPS GTPv2 負荷/過負荷自己保護の除外
- メッセージ優先順位プロファイルで、グローバルレベルまたは各インターフェイスレベル（PFCP、GTP）での優先順位を定義します。

eMPS/WPS サポート

機能説明

この機能は、eMPS サブスクライバの識別をサポートします。この機能は、次のメッセージプライオリティ ビットを設定します。

- UP への PFCP インターフェイス。
- MME および PGW への GTPC インターフェイス。

この機能には、eMPS サブスクライバのプロファイルで構成された値に従って、制御メッセージでの要求メッセージの DSCP マーキングが含まれます。

eMPS GTPv2 負荷/過負荷自己保護除外のサポート

機能説明

cnSGW-C は、eMPS と GTPv2 負荷または過負荷機能との相互作用をサポートします。GTPv2 負荷または過負荷機能での自己保護モード中の eARP コール、APN 緊急コール、または WPS コールの除外をサポートしています。

デフォルトでは、cnSGWc は自己保護モードで着信 GTPC メッセージをスロットルします。cnSGWc が自己保護モードで、過負荷除外プロファイルが構成されている場合、cnSGWc は、着信メッセージのパラメータを構成された DNN、ARP、QCI、およびメッセージの優先順位の値と比較します。

一致した場合、またはセッションが WPS または緊急セッションとして識別された場合、cnSGWc は着信メッセージを処理します。一致しない場合は、そのメッセージを拒否します。

cnSGW-C は、自己保護モードでの着信要求メッセージの拒否から dnn-list、arp-list、および qci-list、message-priority を除外できます。cnSGW-C は、次の方法でこの拒否を除外します。

- サブスクライバ APN 名（NI+OI）が *overload-exclude-profile* と一致する場合に、任意のコールレベルの手順の拒否から dnn-listを除外します。
- ベアラーの変更または作成を、新しいまたは既存の ARP（優先度レベル）値の拒否から除外します。
- ベアラーの変更または作成を、新規または既存の QCI 値の拒否から除外します。
- overload-exclude-profile の構成に関係なく、削除されたベアラーまたはセッション操作（ベアラー削除要求、セッション削除要求、ベアラー削除コマンドなど）を拒否から除外します。

特定のノースバウンド インターフェイスから着信 GTPC トラフィックを除外するには、「過負荷除外プロファイルの構成」の項を参照してください。

（注） cnSGW-C は、メッセージスロットリングをサポートしていません。

メッセージの優先順位プロファイル

表 *78 :* 機能の履歴

機能名	リリース情報	説明
メッセージの優先順位ネゴシエーションを使用した GTPC メッセージの過負荷最適化	2024.04.1	過負荷状態では、cnSGWc は着信メッセージのメッセージの優先順位を定義し、発信メッセージのメッセージの優先順位値をネゴシエーションすることで、スムーズな通信を保証します。 この機能により、ネットワークオペレータは、過負荷除外プロファイルでメッセージの優先順位を設定することで、優先順位が高い着信メッセージをスロットリングから除外できます。 また、この機能は、着信GTPCメッセージのメッセージの優先順位値を設定されたセッションメッセージの優先順位値と比較し、発信 GTPC メッセージの優先順位を最小値に設定します。 **導入されたコマンド**：**message-priority { any | s5e | s11 | sxa } upto** *message_priority* **デフォルト設定**：[Disabled]：有効にするには設定が必要

機能名	リリース情報	説明
メッセージの優先順位プロファイルのサポート	2023.04	この機能により、cnSGWcサービスは次のことができます。 • Priority IPC Stream を使用して、Wireless Priority Service（WPS）セッションのプロトコルポッドにプロセス間通信（IPC）メッセージを送信します。 • メッセージ優先順位プロファイルを作成して、グローバルレベルまたは各インターフェイスレベル（PFCP、GTP）で優先順位を定義します。 **デフォルト設定**：[Disabled]：有効にするには設定が必要。

機能説明

優先順位により、サーバーがサービスリクエストをキューから取り出す順序が決まります。たとえば、クライアントが個々のサービスに割り当てる優先順位は0～15の値であり、0が最高の優先順位を表します。

cnSGW では、cnSGW サービスが Priority IPC Stream を使用して、プロセス間通信（IPC）メッセージを Wireless Priority Service（WPS）セッションのプロトコルポッドに送信します。

cnSGWは、メッセージ優先順位プロファイルを作成して、グローバルレベルまたは各インターフェイスレベル（PFCP、GTP）で優先順位を定義します。メッセージの優先順位値は、以下に基づいて選択できます。

- Create Session Request（CSR）、Create Bearer Response（CBR）、および Create Session Response のベアラーコンテキストで受信した ARP および QCI。
- 更新メッセージの優先順位値は、Update Bearer Request（UBR）のベアラーコンテキストで受信した ARP および QCI に基づきます。

cnSGW は次の機能をサポートしています。

- ローミングパートナーごとの WPS セッション。
- 過負荷スロットリングからの WPS セッションの除外。
- cnSGW でのセッションタイプの競合の解決。

- 併置されたサブスクライバの Sxa インターフェイスを介して UPF で処理する WPS セッション。
- WPS セッションモニタリング。

オペレータは WPS ユーザーの S-GW サービス統計をモニターでき、ユーザーはアクティブな WPS セッションの数もモニターできます。

メッセージ優先順位プロファイルを選択する仕組み

メッセージ優先順位機能

メッセージ優先順位メカニズムの機能は次のとおりです。

- MP（gtp、pfcp）が着信メッセージを処理するために作成されたトランザクションにコピーされ、同じものが子トランザクションにもコピーされます。これらの優先順位の値は、IPC メタデータにもコピーされ、宛先ポッドに送信されます。
- プロトコルポッド（protocol、gtp）は、着信メッセージで使用可能なメッセージの優先順位に基づいて、着信メッセージを WPS メッセージとして検出します。これらのメッセージプロトコルポッドは、サービスポッドに向けて Priority IPC Stream を使用します。

メッセージ優先順位プロファイルの選択のための条件

メッセージ優先順位プロファイルの選択には、次の条件が適用されます。

- サブスクライバポリシーでは、サブスクライバのグループおよび関連するオペレータポリシーを定義する必要があります。
- オペレータポリシーは、関連する DNN ポリシーを定義する必要があります。
- DNN ポリシーは、DNN および関連する DNN プロファイルのリストを定義する必要があります。
- DNN プロファイルには、QoS プロファイル名と WPS プロファイル名が関連付けられている必要があります。
- QoS プロファイルには、message-priority-profile が関連付けられている必要があります。
- message-priority-profile が WPS プロファイルと DNN 用の S-GW QoS プロファイルで設定されている場合、WPS プロファイルで設定された message-priority-profile が優先されます。
- メッセージの優先順位設定が実行時に変更された場合は、新しいコールに適用されます。既存のコールについては、ベアラーが更新されるか、新しいベアラーが追加されたときに新しい構成が適用されます。
- 既存のメッセージの優先順位は、DDN など、セルフトリガーされる発信メッセージに表示されます。
- CSReq、MBReq、DSReq、CBReq、DBReq、UBReq などの PDN レベルの着信 GTP メッセージが有効なメッセージの優先順位とともに受信され、その PDN のいずれかのベアラー

で **copy-incoming** が true に設定されている場合、受信メッセージの優先順位は、その着信メッセージによってトリガーされたすべての発信 GTP/PFCP メッセージにコピーされます。

- HO DSReq、RAB などの UE レベルの着信 GTP メッセージが有効なメッセージの優先順位とともに受信され、その PDN のいずれかのベアラーで **copy-incoming** が true に設定されている場合、受信メッセージの優先順位は、その着信メッセージによってトリガーされた（その PDN の）すべての発信 GTP/PFCP メッセージにコピーされます。
- セッションレポートなどの着信 PFCP メッセージが有効なメッセージの優先順位とともに受信され、その Sx セッションの Sx ベアラーのいずれかで **copy-incoming** が true に設定されている場合、受信メッセージの優先順位が発信応答メッセージにコピーされます。

メッセージの優先順位の選択ロジック

- メッセージの優先度の値は、次のロジックに基づいて発信メッセージに表示されます。
  - **GTP メッセージ**：
    - **PDN レベルのメッセージの場合**：PDN に WPS としてマークされた複数の（GTP メッセージの優先順位が異なる）ベアラーがある場合、優先順位が最も高い（0 が最高、15 が最低）メッセージが発信 GTP メッセージにコピーされます。
    - **UE レベルの GTP メッセージの場合**：UE に WPS としてマークされた複数の（GTP メッセージの優先順位が異なる）ベアラーがある場合、優先順位が最も高い（0 が最高、15 が最低）メッセージが発信 GTP メッセージにコピーされます。
  - **PFCP メッセージの場合**：Sx セッションに WPS としてマークされた複数の（GTP メッセージの優先順位が異なる）ベアラーがある場合、優先順位が最も高い（0 が最高、15 が最低）メッセージが発信 PFCP メッセージにコピーされます。

発信 GTPC メッセージに関するメッセージの優先順位ネゴシエーション

発信 GTPC メッセージの場合、cnSGWc は着信 GTPC メッセージで受信したメッセージの優先順位値と設定に基づいて決まるメッセージの優先順位値を比較し、最も小さい値を選択します。このメッセージの優先順位値は、発信 GTPC 要求または応答メッセージに割り当てられます。

cnSGW でのセッションタイプの競合の解決

cnSGW は、次のタグを付けてセッションタイプの競合を解決します。

- DNN に緊急タグが付いている場合、または作成されたセッションが緊急セッションの場合、セッションは緊急としてタグ付けされます。緊急セッションでは、cnSGW は CS 要求で受信した QCI/ARP を使用してメッセージ優先順位を選択します。
- DNN プロファイルで IMS/VOLTE として設定された QCI を使用してベアラーが作成された場合、セッションは IMS/VOLTE としてタグ付けされます。
- 受信した QCI/ARP に基づいて、セッションは WPS としてタグ付けされます。

- セッションが複数の条件を満たしている場合、セッションタイプは次の基準に基づいてタグ付けされます。

  WPS > 緊急 > IMS

Sxa インターフェイスを介した UPF での WPS セッションの処理

CnSGW は、次の方法で Sxa インターフェイスを介した UPF での WPS セッションを処理します。

- message-priority プロファイルで PFCP インターフェイスに定義されているメッセージの優先順位を選択します。たとえば、[Priority] 値 0 は最も高い優先順位と見なされ、15 は最も低い優先順位と見なされます。
- この優先順位は、PFCP ヘッダーで UPF に渡されます。
- UPF は、WPS セッションの早期リカバリのために、セッションリカバリ中にこの優先順位を使用します。

（注） 併置されたサブスクライバの場合、UPF は、cnSGW と SMF がこれらのセッションを同時にリカバリするために同じメッセージの優先順位を選択することを想定しています。

- PDN の 1 つが WPS で他の PDN が非 WPS である multipdn コールがある場合、cnSGW は WPS PDN に対してのみ WPS の優先順位を選択します。
- 併置されたコールがある場合、cnSGW は WPS の優先順位を選択し、SMF はセッションに対して WPS 以外の優先順位を選択します。このような場合、UPF は、併置されたセッションのリカバリに最も高い優先順位を使用します。このようにして、UPF は、両方のセッションが同時に回復するようにします。

WPS セッションモニタリング

セッションが WPS であるかどうかをモニタリングするには、S-GW サービス統計を使用します。「緊急」、「IMS/VOLTE」、「WPS」などのセッションタイプの値を使用して、セッションにタグを付けることができます。WPS ベアラーが削除されると WPS タギングが削除され、セッションがアクティブな間、正常から WPS に、または WPS から正常に変わります。セッションが WPS に変換されると、新しい非一意のキーがセッションに追加され、WPS 以外に戻ると削除されます。これは、**show subscriber wps** および **clear subscriber wps** に使用されます。cnSGWc は **show subscriber count wps** をサポートしています。詳細については、「モニタリングとトラブルシューティング」の項を参照してください。

機能設定

この機能の設定には、次の手順が含まれます。

- WPS プロファイルを設定します。詳細については、WPS プロファイルの設定 （233 ページ） を参照してください。
- メッセージの優先順位プロファイルを設定します。詳細については、メッセージの優先順位プロファイルの設定 （234 ページ）を参照してください。
- SGW プロファイルを設定し、WPS プロファイルと SGW プロファイルの関連付けを有効にします。詳細については、WPS プロファイルと SGW プロファイルの関連付けの構成（235 ページ） を参照してください。
- S-GW QoS プロファイルを設定します。詳細については、SGW QoS プロファイルの構成（236 ページ）を参照してください。
- DNN プロファイルを設定し、WPS プロファイルと DNN プロファイルの関連付けを有効にします。詳細については、WPS プロファイルと DNN プロファイルの関連付けの設定（236 ページ） を参照してください。
- SGW QoS プロファイルを SGW プロファイルおよび DNN プロファイルに関連付けます。詳細については、sgw-qos-profile を sgw-profile および DNN プロファイルと関連付け （237 ページ）を参照してください。

WPS プロファイルの設定

この機能を設定するには、次の構成を使用します。

```
config
  profile wps wps_name
    arp arp_value message-priority-profile msg_priority_profile_name
    dscp dscp_value
    message-priority [ pfcp | gtpc ]
    end
```

注：

- **wps** *wps_name*：WPS サービス名を指定します。文字列で指定する必要があります。
- **arp** *arp_value*：ARP レベルの範囲を指定します（, または - で区切ります）。整数または文字列で指定する必要があります。WPS セッションは ARP に基づいて決定されます。
- **message-priority-profile** *msg_priority_profile_name*：メッセージ優先順位プロファイルが WPS プロファイル内の ARP リストに追加されるように指定します。WPS セッションは、WPS プロファイル内で設定されている ARP および関連付けられたメッセージ優先順位プロファイルに基づいて決定されます。

- **dscp** *dscp_value*：DSCP マーキング値を 10 進数（0 ～ 63 の範囲）または 16 進数（0x0 ～ 0x3F の範囲）で指定します。文字列で指定する必要があります。

設定例

次に、設定例を示します。

```
config
   profile wps wp1
     arp 2 message-priority-profile message_priority_name
   end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show full-configuration profile wps wps1
profile wps wps1
arp 2 message-priority-profile mp1
exit
```

メッセージの優先順位プロファイルの設定

メッセージの優先順位プロファイルを設定するには、次の設定を使用します。

```
config
  profile  message-priority msg_priority_profile_name
     interface [ any | gtp | pfcp | sbi ] priority [ value range_value |
 copy-incoming ]priority copy_incoming_value
  end
```

注：

- **profile message-priority-profile***msg_priority_profile_name*：メッセージの優先順位プロファイルとプロファイル名を指定します。
- **interface [any | gtp | pfcp | sbi]** ：次のいずれかのインターフェイスを指定します。
  - any
  - gtp
  - pfcp
  - sbi
- **Interface gtp priority [value** *range_value* ：優先順位値を指定します。
  - 0 ー 31 の範囲の値は SBI インターフェイス用で、0 ー 15 はその他のインターフェイス用です。
  - 優先順位値の範囲が 0 ～ 15 の場合、0 が最も高い優先順位、15 が最も低い優先順位を表します。

- 優先順位値は、インターフェイスごとに設定できます。インターフェイスタイプの設定は任意で、設定されていない場合、すべてのインターフェイスに同じ値が適用されます。

- **Interface gtp priority [value** *range_value* **| copy-incoming]***priority_copy_incoming_value*：copy-incomingが設定されている場合、cnSGWは、着信メッセージで受信した優先順位を、その着信メッセージによってトリガーされたすべての発信メッセージにコピーします。

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show full-configuration profile message-priority mp1
profile message-priority mp1
interface gtp priority value 2
interface gtp priority copy-incoming
exit
```

WPS プロファイルと SGW プロファイルの関連付けの構成

WPSプロファイルとSGWプロファイルの関連付けを構成するには、次の構成を使用します。

```
config
 profile sgw sgw_name
  wps-profile wps_name
  end
```

注：

- **wps-profile** *wps_name*：ワイヤレス優先サービス（WPS）名を指定します。文字列で指定する必要があります。

設定例

次に、設定例を示します。

```
config
   profile sgw sgw1
     wps-profile wp1
     end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw
profile sgw sgw1
wps-profile wp1
```

WPS プロファイルと DNN プロファイルの関連付けの設定

ここでは、WPS プロファイルと DNN プロファイルの関連付けの設定方法について説明します。

（注） WPS プロファイルが SGW プロファイルおよび DNN プロファイルに関連付けられている場合、DNN プロファイルが優先されます。

WPS プロファイルと DNN プロファイルの関連付けを設定するには、次の設定を使用します。

```
config
 profile dnn dnn_name
   wps-profile wps_name
   end
```

設定例

次に、設定例を示します。

```
config
  profile dnn dnn1
  wps-profile wps1
  end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile dnn
profile dnn dnn1
wps-profile wps1
```

SGW QoS プロファイルの構成

メッセージの優先順位プロファイルを、SGW QoS プロファイルと、SGW QoS プロファイルの {qci,arp} に関連付けることができます。{qci,arp} に関連付けられたメッセージの優先順位プロファイルと qos プロファイルに関連付けられたメッセージの優先順位プロファイル、および WPS に関連付けられたメッセージの優先順位プロファイルをネゴシエートして、より高い優先順位を持つプロファイルを見つけます。

```
config
  profile sgw-qos-profile  qos_profile_name
     message-priority-profile message_priority_profile_name
     qos qci nonstandard_value arp-priority-level arp_value [
message-priority-profile msg_priority_profile ]
  exit
```

注：

- **message-priorty-profile** *message_priority_profile_name*：メッセージの優先順位プロファイル名を構成します

- **qos qci** *standard_value* **arp-priority-level** *arp_value* ：QCI/ARP 値の QoS パラメータを構成します。1 ～ 254 の範囲の整数で指定する必要があります。たとえば、範囲の値は 1-9、65、66、69、70、80、82、83、128-254 です。
- **arp-priority-level** *arp_value* **[message-priority-profile** *msg_priority_profile* **]**：ARP 優先度レベル（1 ～ 15）を構成します。

設定例

次に、設定例を示します。

```
config
    profile sgw-qos-profile sgw1
      message-priority-profile mp1
      qos qci 2 arp-priority-level 2 message-priority-profile mp1
```

構成の確認

構成を確認します。

```
smf(config)# show full-configuration profile sgw-qos-profile sgw1
profile sgw-qos-profile sgw1
message-priority-profile mp1
qos qci 2 arp-priority-level 2 message-priority-profile mp1
exit
```

sgw-qos-profile を sgw-profile および DNN プロファイルと関連付け

次の構成例を使用して、WPS プロファイルを構成済みの DNN プロファイルに関連付けます。

```
config
  profile dnn profile_dnn_name
    wps-profile wps_profile_name
    qci-qos-profile    sgw_qos
    end
```

注：

- **wps-profile** *wps_profile_name*：wps プロファイル構成を有効にします。このプロファイルは、既存の DNN プロファイル構成で構成されます。
- **qci-qos-profile** *qos_profile_name*：S-GW の QoS プロファイル構成名を指定します。

次の構成例を使用して、sgw-qos-profile を構成済みの sgw プロファイルに関連付けます。

```
config
  profile sgw sgw_name
  qci-qos-profile    sgw_qos
  end
```

注：

- **qci-qos-profile** *qos_profile_name*：S-GW の QoS プロファイル構成名を指定します。

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

モニタリングおよびトラブルシューティング

このセクションでは、WPS セッションモニタリングについて説明します。詳細を表示するには、次の show sub CLI で次のサブタイプラベルを使用します。

show subscriber namespace sgw sub-type *subscriber type*

```
show subscriber nf-service sgw sub-type wps gr-instance 1
Wed Aug  9  09:20:49.500 UTC+00:00
subscriber-details
{
  "subResponses": [
    [
      "id-index:1:0:32768",
      "id-index-key:1:0:globalKey:32768",
      "id-value:16779392",
      "imsi:imsi-123456789012348",
      "msisdn:msisdn-223310101010101",
      "imei:imei-123456786666660",
      "upf:10.1.11.32",
      "upfEpKey:10.1.11.32:10.1.14.119",
      "subscribertype:wps",
      "s5s8Ipv4:10.1.14.19",
      "s11Ipv4:10.1.11.32",
      "namespace:sgw",
      "nf-service:sgw"
    ]
  ]
}
```

show subscriber namespace sgw count sub-type *subscriber type*

指定されたサブスクライバタイプ、SUPI 値のカウントの詳細を表示できます。

```
[sgw] smf# show subscriber namespace sgw count su
Possible completions:
  sub-type
  supi
[sgw] smf# show subscriber nf-service sgw count sub-type wps
Wed Aug  9  09:20:53.760 UTC+00:00
subscriber-details
{
  "sessionCount": 1
}
```

clear subscriber namespace sgw sub-type *subscriber type*

```
[sgw] smf# clear subscriber namespace sgw sub-type ?
Description: Specify Subscriber Type [wps|emergency|volte|non-volte]
Possible completions:
  <string>
[sgw] smf# clear subscriber namespace sgw sub-type wps
Wed Jul  19 10:30:07.716 UTC+00:00
```

```
result
ClearSubscriber Request submitted
```

show sessions summary slice-name *slice_name*

```
[sgw] smf# cdl show sessions summary slice-name 1
Wed Aug  9  09:21:08.552 UTC+00:00
session {
    primary-key 2#/#imsi-123456789012348
    unique-keys [ "2#/#16779392" ]
    non-unique-keys [ "2#/#id-index:1:0:32768" "2#/#id-index-key:1:0:globalKey:32768"
"2#/#id-value:16779392" "2#/#imsi:imsi-123456789012348" "2#/#msisdn:msisdn-223310101010101"
 "2#/#imei:imei-123456786666660" "2#/#upf:10.1.11.32" "2#/#upfEpKey:10.1.11.32:10.1.14.119"
 "2#/#subscribertype:wps" "2#/#s5s8Ipv4:10.1.14.19" "2#/#s11Ipv4:10.1.11.32"
"2#/#namespace:sgw" ]
    flags [ flag3:a010b20:a010b20,a010b20:a010e13,
byte-flag1:02:13:03:53:00:00:08:16:0A:01:0B:20:11:FF:01:5B:18:21:63:54:09:2A:21:63:54:00:12:D6:87:10:01:21:63:54:00:00:01:89:D9:99:3F:4E
 session-state-flag:sgw_active ]
    map-id 1
    instance-id 1
    version 1
    create-time 2023-08-09 09:20:46.302786264 +0000 UTC
    last-updated-time 2023-08-09 09:20:46.867525921 +0000 UTC
    purge-on-eval false
    next-eval-time 2023-08-16 09:20:46 +0000 UTC
    session-types [ SGW:rat_type:EUTRAN SGW:colocated:false SGW:pdn_active:1
SGW:bearer_active:1 SGW:subscriber_type:wps SGW:apn:intershat ]
    data-size 950
}
```

バルク統計サポート

次に、eMPS メッセージの例を示します。

```
sgw_pdn_emps_counters{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
 service_name="sgw-service",status="active"} 1
```

```
sgw_pdn_emps_stats{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
 service_name="sgw-service",status="release"} 7
```

```
sgw_pdn_emps_stats{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
 service_name="sgw-service",status="setup"} 8
```

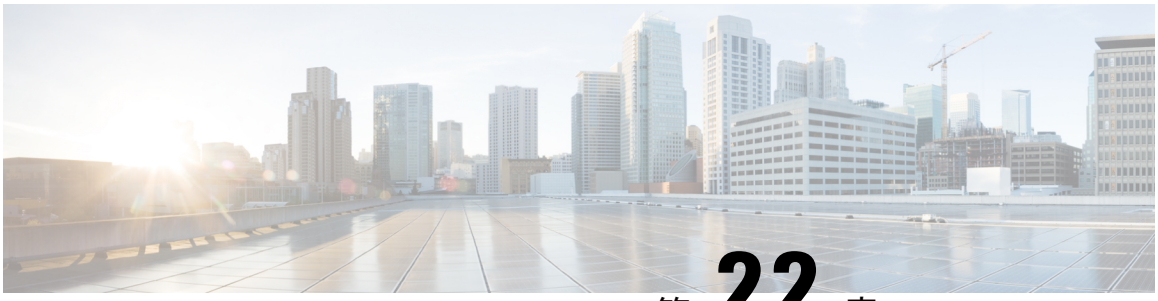
```
gtpc_app_priority_events{app_name="smf",cluster="smf",data_center="smf",event_type=
 "NumRxModifyBearerResFrmSerSuccess",instance_id="0",
interface_type="S11",priority_msg="true",service_name="gtpc-ep"} 3
```

```
gtpc_app_priority_events{app_name="smf",cluster="smf",data_center="smf",event_type=
 "RxCreateSessionRes",instance_id="0",interface_type="S5E",
priority_msg="true",service_name="gtpc-ep"} 2
```

```
proto_pfcp_msg_total{app_name="smf",cluster="smf",data_center="smf",instance_id="0",
 interface_type="SXA",message_direction="outbound",
```

```
message_name="N4_MSG_SESSION_ESTABLISHMENT_REQUEST",msgpriority="True",service_name=
 "protocol",status="accepted",transport_type="origin"} 2
```

```
proto_pfcp_msg_total{app_name="smf",cluster="smf",data_center="smf",instance_id="0",
 interface_type="SXA",message_direction="outbound",
message_name="N4_MSG_SESSION_MODIFICATION_REQUEST",msgpriority="True",service_name="protocol",
 status="accepted",transport_type="origin"} 6
```



第 22 章

障害およびエラーの処理サポート

- 機能の概要と変更履歴 （241 ページ）
- 概要 （242 ページ）
- 接続と接続解除の失敗とエラーの処理 （242 ページ）
- ベアラー作成/更新/削除要求および応答の失敗とエラーの処理 （246 ページ）
- 無線アクセスベアラーや Modify Bearer Request の障害とエラーの処理 （252 ページ）
- 障害原因コード、原因ソース、ベアラーコンテキストエラーのサポート （254 ページ）

機能の概要と変更履歴

要約データ

表 *79*: 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *80 :* マニュアルの変更履歴

改訂の詳細	リリース
情報要素（IE）のサポートが追加されました。 • 障害原因コード • 原因ソース（ビット 1：CS） • ベアラーコンテキスト IE エラー（ビット 2：BCE）	2021.02.3
最初の導入。	2021.01.0

概要

cnSGW-C は以下をサポートします。

- 接続と接続解除の失敗とエラーの処理。
- ベアラー作成、更新、削除要求および応答の失敗とエラーの処理。
- 無線アクセスベアラーまたはベアラー変更要求の失敗とエラーの処理。

コールの処理中に発生する可能性のあるさまざまなタイプの障害を次に示します（セッション セットアップ タイマーを除く）。

- 要求と応答の事前検証での失敗。
- 再送信タイムアウト。
- トランザクション サービス レベル契約（SLA）
- ピアから報告された障害（メッセージプロセスの段階に応じてUP、PGW、またはMME）。

接続手順中のセッションセットアップタイマーでは、次の障害が発生する可能性があります。

- 初期接続およびマルチ PDN で、進行中の PDN の確立と MME からのベアラー変更要求が受信されない。

接続と接続解除の失敗とエラーの処理

cnSGW-C は次をサポートしています。

- タイムアウト機能の設定

- 次の失敗応答の処理：
  - 初期接続および追加の PDN 設定手順の一部としての Clear Session Request
  - PGW および UPF に対する Delete Beare Request および Delete Session Request 処理

Create Session Request の障害処理

ここでは、Create Session Request 手順の障害シナリオについて説明します。

初期接続手順中にエラーが発生した場合、サブスクライバコンテキストは作成されません。

別の PDN のセットアップが失敗した場合、PDN はサブスクライバコンテキストで作成されません。

次の表に、さまざまな障害タイプに対するコール処理の各段階における cnSGW-C の動作の一覧を示します。

表 *81 : Create Session Request* 手順の障害シナリオにおける *cnSGW-C* の動作

シナリオ	Sx シグナリング（復元）	S11 シグナリング（復元）	S5 シグナリング
• Create Session Request の事前検証の失敗 • ユーザープレーン（UP）からの Sx Session Establishment Response 障害	なし	負：Create Session Response	なし
• Create Session Response 障害	対応 新たに作成されたベアラーのトラフィックエンドポイントを削除します。	負：Create Session Response	なし
• Sx Modify Response プロセスの失敗	対応 新たに作成されたベアラーのトラフィックエンドポイントを削除します。	負：Create Session Response	新たに作成されたセッションを削除する Delete Session Request

デフォルトのベアラーの削除手順の失敗処理

このセクションでは、PGW によって開始されるデフォルトのベアラーの削除手順の失敗のシナリオについて説明します。

次の表に、さまざまな障害タイプに対するコール処理の各段階における cnSGW-C の動作の一覧を示します。

表 *82 :* デフォルトのベアラーの削除手順の失敗シナリオにおける *cnSGW-C* の動作

シナリオ	SGW サービスの動作	Signaling	出力
ベアラーの削除要求の事前検証の失敗	送信失敗/Sx/PGW 経由のシグナリングなし	否定応答：S5 ベアラー削除応答	セッション/PDN は削除されませんでした
Sx セッション変更失敗	失敗をスキップして続行	S11 ベアラー要求を MME に送信 S5 ベアラー削除要求（原因 = Accepted） Sx セッション変更要求に応じた Sx 削除要求（トラフィックエンドポイントを削除するため）	セッション/PDN が削除されました
S11 ベアラー要求の失敗	失敗をスキップして続行	ベアラー削除応答（原因 = Accepted）を PGW に送信 Sx 削除要求（トラフィックエンドポイントを削除するため）	セッション/PDN が削除されました
Sx セッション削除要求の失敗	失敗をスキップして続行	ベアラー削除応答（原因 = Accepted）を PGW に送信	セッション/PDN が削除されました

セッション削除手順の障害処理

ここでは、MME が開始したセッション削除手順の障害シナリオについて説明します。

次の表に、障害シナリオにおける cnSGW-C の動作を示します。

表 *83* : セッション削除手順の障害シナリオにおける *cnSGW-C* の動作

シナリオ	SGW サービスの動作	Signaling	出力
Delete Session Request の事前検証の失敗	送信失敗または Sx/MME 経由のシグナリングなし	負：S11 Delete Session Response	セッション/PDN は削除されませんでした
Sx Session Modification Request の失敗	失敗をスキップして続行	Delete Session Request を PGW に送信 Delete Session Response（原因 = Accepted）を MME に送信 Sx Session Modify Response に応じた Sx Delete Request（トラフィックエンドポイントを削除するため）	セッション/PDN が削除されました
S5 Delete Session Request の失敗	失敗をスキップして続行	Delete Session Response（原因 = Accepted）を MME に送信 Sx Delete Request（トラフィックエンドポイントを削除するため）	セッション/PDN が削除されました
Sx Session Deletion Request の失敗	失敗をスキップして続行	Delete Session Response（原因 = Accepted）を MME に送信	セッション/PDN が削除されました

接続手順中のセッション セットアップ タイマー

ここでは、接続手順中のセッション セットアップ タイマーについて説明します。

次の表に、セッション タイムアウト シナリオでの cnSGW-C の動作を示します。

表 *84* : セッションタイムアウト障害シナリオでの *cnSGW-C* の動作

シナリオ	Sx シグナリング（復元）	S11 シグナリング（復元）	S5 シグナリング	出力
SGW がセッション作成応答を MME に送信し、MME からのベアラー変更要求を待った後、セッションタイムアウト設定が期限切れになりました。	Sx 削除要求（新たに作成されたベアラーのトラフィックエンドポイントを削除するため）	セッション削除要求	ベアラー削除要求	セッション/PDN が削除されました

ベアラー作成/更新/削除要求および応答の失敗とエラーの処理

このセクションでは、ベアラー作成/更新/削除要求および応答の失敗とエラーの処理シナリオについて説明します。

ベアラー作成手順の失敗の処理

このセクションでは、PGW によって開始される専用ベアラー作成手順の失敗のシナリオについて説明します。

次の表に、さまざまな障害タイプに対するコール処理の各段階における cnSGW-C の動作の一覧を示します。

（注） 専用ベアラー作成要求および応答の処理中に、SGW がピアから「Context Not Found」（MME/UP）を受信すると、この原因を送信したピアに対するシグナリングを何も実行せずに PDN を削除します。

表 *85 :* ベアラー作成手順の失敗シナリオでの *cnSGW-C* の動作

シナリオ	**Sx** シグナリング（リバート処理）	**S11** シグナリング（リバート処理）	**S5** シグナリング	出力
• ベアラー作成要求の事前検証に失敗しました • Sx セッション変更要求が失敗しました（トンネルエンドポイントと GTPU TEID を割り当てるために UP に送信された要求）。	非対応	非対応	否定：PGW へのベアラー作成応答	新しいベアラーコンテキストは作成されません。

シナリオ	Sx シグナリング（リバート処理）	S11 シグナリング（リバート処理）	S5 シグナリング	出力
ベアラー作成要求が MME に送信され、SGW は応答を待っています。	対応 新しく作成されたベアラーのトラフィックエンドポイントを削除します。	なし	否定：PGW へのベアラー作成応答	新しいベアラーコンテキストは作成されません。 ベアラー作成応答のオプションのパラメータは無視されます。
SGW は MME からベアラー作成応答を受信し、Sx Modify を UP に送信して eNodeB と SGW-U の間の GTPU トンネルを接続します。	対応 新しく作成されたベアラーのトラフィックエンドポイントを削除します。	対応 新しく作成されたベアラーを削除します。	否定：PGW へのベアラー作成応答	新しいベアラーコンテキストは作成されません。 ベアラー作成応答のオプションのパラメータは無視されます。
ベアラー削除要求または Sx 変更要求のリバート処理に失敗しました。	非対応	非対応	否定：ベアラー作成応答	ベアラーコンテキストは作成されません。 ベアラー作成応答のオプションのパラメータは無視されます。

削除専用ベアラー手順の失敗処理

このセクションでは、PGW によって開始される専用のベアラーの削除手順の失敗のシナリオについて説明します。

次の表に、さまざまな障害タイプに対するコール処理の各段階における cnSGW-C の動作の一覧を示します。

（注） 専用ベアラー削除要求および応答の処理中に、SGW がピアから「Context Not Found」（MME/UP）を受信すると、この原因を送信したピアに対するシグナリングを何も実行せずに PDN を削除します。

表 *86 :* 専用ベアラーの削除手順の失敗シナリオにおける *cnSGW-C* の動作

シナリオ	SGW サービスの動作	Signaling	出力
ベアラーの削除要求の事前検証の失敗	送信失敗または Sx/MME 経由のシグナリングなし	否定：PGW へのベアラー削除応答	専用ベアラーは削除されません。
Partially Accepted：複数の EBI を含むベアラー削除要求が受信された場合。これは次のような場合です。 • 一部の EBI が PDN に属している • 一部の EBI が PDN に属していないか、EBI が無効である	DBR 手順を続行し、ベアラーの除要求を受信したすべての既存のベアラーを削除します。 S11 ベアラー削除要求は既存の EBI 情報のみを伝送する必要があります。 Sx セッション変更要求は、既存の EBI 情報を伝送する必要があります（トラフィックエンドポイントの削除）	S5 ベアラー削除応答（原因 = Partially Accepted）	メッセージレベルの原因が一部承認され、ベアラレベルの原因が次の場合の S5 ベアラー削除応答。 • **一部の EBI が PDN に属している**： S11 ベアラー削除応答 • **一部の EBI が PDN に属していない/無効な EBI**：Context Not Found CDL が更新されます（すべての既存のベアラーを削除）
アクションを Drop に設定するための Sx セッション変更	失敗をスキップして続行	S5 ベアラー削除応答（原因 = Accepted）	失敗をスキップして DBR プロシージャコールフローを続行：はい S5 ベアラー削除応答（原因 = Accepted） CDL が更新されます。
S11 ベアラー要求の失敗	失敗をスキップして続行	S5 ベアラー削除応答（原因 = Accepted）	失敗をスキップして DBR プロシージャコールフローを続行：はい CDL が更新されます。
Sx セッション変更要求の失敗（UP のトラフィックエンドポイントの削除要求）	失敗をスキップして続行	S5 ベアラー削除応答（原因 = Accepted）	失敗をスキップして DBR プロシージャコールフローを続行：はい CDL が更新されます。

Update Bearer 手順の障害処理

ここでは、PGW が開始した Update Bearer 手順の障害シナリオについて説明します。

次の表に、さまざまな障害タイプに対するコール処理の各段階における cnSGW-C の動作の一覧を示します。

表 *87 : Update Bearer* 手順の障害シナリオにおける *cnSGW-C* の動作

シナリオ	S5/Sx シグナリング	出力
Update Bearer Request の事前検証の失敗	負：Update Bearer Response	ベアラー/PDN コンテキストに変更はありません。
存在しない EBI を使用した Update Bearer Request	メッセージレベルの理由が REQ_PARTIALLY_ACCEPTED である Update Bearer Response。存在しない EBI のベアラーレベルの原因（Context Not Found）。（既存の EBI の通常の処理）	メッセージレベルの理由が REQ_PARTIALLY_ACCEPTED である Update Bearer Response。 存在しない EBI のベアラーレベルの原因（CONTEXT_NOT_FOUND）。 既存の EBI に対する通常の処理。 CDL は既存の EBI についてのみ更新されます。
Update Bearer Request が MME に送信され、応答を待機している	負：Update Bearer Response	負：Update Bearer Response CDL は更新されません。
S11 Update Bearer Response（メッセージレベルの原因 == CONTEXT_NOT_FOUND）および S5 Update Bearer Req/Rsp に、ベアラーコンテキストリスト内のデフォルトのベアラーが存在している	Sx Delete Req/Rsp 負：Update Bearer Response	負：Update Bearer Response CDL は更新されません。 統計/トランザクションログ PDN キーのリリースと PDN の割り当て解除 これが最後の PDN の場合は、リソースマネージャ、およびすべてのサブスクライバキーがリリースされ、サブスクライバの割り当てが解除されます。
S11 Update Bearer Response（メッセージレベルの原因 == CONTEXT_NOT_FOUND）および S5 Update Bearer Req/Rsp に、ベアラーコンテキストリスト内のデフォルトのベアラーが存在していない	Sx Modify Req/Rsp 負：Update Bearer Response	CDL は更新されません。

シナリオ	S5/Sx シグナリング	出力
S11 Update Bearer Response（メッセージレベルの原因 == REQ_PARTIALLY_ACCEPTED、ベアラーコンテキストの原因==専用ベアラーの障害）	Sx Modify Req/Rsp メッセージレベルの理由が REQ_PARTIALLY_ACCEPTED である Update Bearer Response	メッセージレベルの理由が REQ_PARTIALLY_ACCEPTED である Update Bearer Response 成功したベアラーに対する CDL が更新されます。
S11 Update Bearer Response（メッセージレベルの原因 == REQ_PARTIALLY_ACCEPTED、ベアラーコンテキストの原因==デフォルトベアラーの CONTEXT_NOT_FOUND）	Sx Delete Req/Rsp 負：Update Bearer Response	負：Update Bearer Response CDL は更新されません。 PDN キーのリリースと PDN の割り当て解除 これが最後の PDN の場合は、リソースマネージャ、およびすべてのサブスクライバキーがリリースされ、サブスクライバの割り当てが解除されます。
Update Bearer Response 後に Sx 変更がトリガーされた場合： • Sx Session Modification Request（IPC/再送信タイムアウト/内部障害など） • Sx Session Modify Response（原因 != CONTEXT_NOT_FOUND を除く ACCEPTED）	障害を無視して続行	障害を無視して続行
Update Bearer Response 後に Sx 変更がトリガーされた場合： • Sx Session Modify Response（原因 == CONTEXT_NOT_FOUND）	[1]	[2]

シナリオ	S5/Sx シグナリング	出力
Update Bearer Response 後に Sx 削除がトリガーされた場合： • Sx Session Delete Request（IPC/再送信タイムアウト/内部障害など） • Sx Delete Session Response（原因 != ACCEPTED） • リソースマネージャのリリース（内部エラー）	障害を無視して続行	障害を無視して続行

[1] Update Bearer 手順の一環として、SGW は PDN の削除用に新しいトランザクションをトリガーします。

- Sx Session Modification Response の一環として Sx Failure Cause を受信
  - Context Not Found

SGW の動作（新規トランザクション）：

- SGW が、その PDN を削除するために S11 Delete Bearer Request および S5 Delete Session Request をトリガーします。
- No Sx Signaling

SGW の動作（Update Bearer トランザクション）：SGW が、Update Bearer 手順トランザクションの一環として、No Resource Available の理由で S5 Update Bearer Response を送信します。また、SGW は、Context Not Found の理由で Sx Session Modification Response を受信してもすぐには UP へのシグナリングを開始しません。

[2] Update Bearer 手順の一環として、SGW-SVC は PDN の削除用に新しいトランザクションをさらにトリガーします。

- Sx Session Modification Response の一環として Sx Failure Cause を受信
  - Context Not Found

SGW の動作（新規トランザクション）：

- SGW が、その PDN を削除するために S11 Delete Bearer Request および S5 Delete Session Request をトリガーします。
- No Sx Signaling

SGW の動作（Update Bearer トランザクション）：SGW が、Update Bearer 手順トランザクションの一環として、No Resource Available の理由で S5 Update Bearer Response を送信

します。また、SGW は、Context Not Found の理由で Sx Session Modification Response を受信してもすぐには UP へのシグナリングを開始しません。

無線アクセスベアラーや Modify Bearer Request の障害とエラーの処理

ここでは、無線アクセスベアラー（RAB）、PGW からの Modify Bearer Request および Response（MBR）、およびユーザープレーン（UP）の障害シナリオについて説明します。

次の表に、さまざまな障害タイプに対するコール処理の各段階における cnSGW-C の動作の一覧を示します。

表 *88 :* 無線アクセスベアラーと *Modify Bearer Response* 手順の障害シナリオにおける *cnSGW-C* の動作

メッセージタイプ	障害インターフェイス	受信した失敗応答	送信される失敗応答	ハンドル
MBR 初期接続	Sx	CONTEXT_NOT_FOUND	EGTP_CAUSE_NO_RESOURCES_AVAILABLE	PGW への DSR および MME への DBR で PDN をクリーンアップする
		その他の失敗応答	EGTP_CAUSE_NO_RESOURCES_AVAILABLE	PGW への DSR および MME への DBR で PDN をクリーンアップする Sx_Modification_Req/ Sx_Session_Delete で UP のリソースをクリーンアップする
	タイムアウト（Timeout）	PFCP でのタイムアウト	EGTP_CAUSE_NO_RESOURCES_AVAILABLE	PGW への DSR および MME への DBR で PDN をクリーンアップする Sx_Modification_Req/ Sx_Session_Delete で UP のリソースをクリーンアップする

メッセージタイプ	障害インターフェイス	受信した失敗応答	送信される失敗応答	ハンドル
MBR サービスリクエスト	Sx	CONTEXT_NOT_FOUND	EGTP_CAUSE_ NO_RESOURCES_ AVAILABLE	PGW への DSR および MME への DBR で PDN をクリーンアップする
		その他の失敗応答	EGTP_CAUSE_ NO_RESOURCES_ AVAILABLE	PDN 内では何も更新せず、S5 シグナリングを無視する
		タイムアウト（Timeout）	EGTP_CAUSE_ NO_RESOURCES_ AVAILABLE	PDN 内では何も更新せず、S5 シグナリングを無視する
	S5	EGTP_CAUSE_ CONTEXT_NOT_ FOUND	EGTP_CAUSE_ CONTEXT_NOT_ FOUND	Sx_Session_Delete を UP に送信する MME への MBRsp 障害（DBR/DSR なし）
		その他の失敗応答	PGW から受信した失敗応答	PDN/DB を更新しない
	タイムアウト（Timeout）	PGW からのタイムアウト	EGTP_CAUSE_ PEER_NOT_ RESPONDING	PDN/DB を更新しない

メッセージタイプ	障害インターフェイス	受信した失敗応答	送信される失敗応答	ハンドル
RAB	Sx	CONTEXT_ NOT_FOUND （シングル PDN コール）	EGTP_CAUSE_ REQ_ACCEPTED	PGW への DSR および MME への DBR で PDN をクリーンアップする
		CONTEXT_ NOT_FOUND （マルチ PDN コールとコンテキストが 1 つの PDN に対して見つからない）	RAB 応答の送信 （EGTP_CAUSE_ REQ_ACCEPTED）	PGW への DSR および MME への DBR で受信したコンテキストが見つからない PDN をクリーンアップする 他の PDN/UE を IDLE に移動する
		その他の失敗応答	EGTP_CAUSE_ REQ_ACCEPTED	PDN/UE を IDLE に移動する
		タイムアウト （Timeout）	EGTP_CAUSE_ REQ_ACCEPTED	PDN/UE を IDLE に移動する

障害原因コード、原因ソース、ベアラーコンテキストエラーのサポート

ここでは、情報要素（IE）：障害原因コード、原因ソース（CS）、およびベアラーコンテキストエラー（BCE）のサポートについて説明します。

IE の技術仕様の詳細については、3GPP TS 29.274 を参照してください。

障害原因コード

cnSGW-C は、S5 インターフェイスで SMF から受信した障害原因コードを処理します。

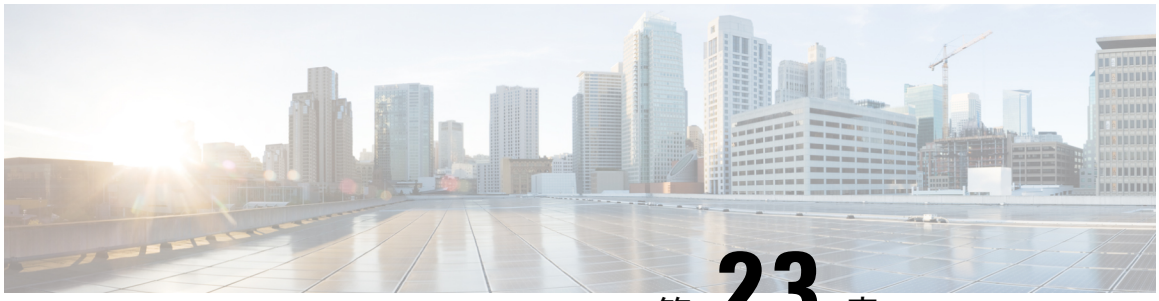
原因ソース

ビット 1：CS（原因のソース）値「0」は、メッセージを送信しているノードによってエラーが発信されたことを示します。

ビット1：CS（原因のソース）の値「1」は、エラーがリモートノード（MME/SGSNからPGWへ、またはPGWからMME/SGNへ）によって発信されたことを示します。

ベアラーコンテキストエラー

ビット2-BCE（ベアラーコンテキストIEエラー）のデフォルト値は「0」です。BCEビットが1の場合は、対応する拒否句がベアラーコンテキストIEのエラーが原因であることを示します。原因値が受け入れ原因値の1つである場合、このビットは廃棄されます。



第 23 章

GTPC および Sx パスの管理

- 機能の概要と変更履歴 （257 ページ）
- 機能説明 （258 ページ）
- GTPC および Sx パスの管理 （258 ページ）
- GTPC パス障害 （264 ページ）
- Sx パス障害 （267 ページ）
- パス障害検出のカスタマイズ （269 ページ）

機能の概要と変更履歴

要約データ

表 *89* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	GTPC および Sx パス管理：無効 - 有効にするには構成が必要 GTPC パス障害：有効 - 常時オン Sx パス障害：有効 - 常時オン パス障害検出のカスタマイズ：無効 - 有効にするには構成が必要
関連資料	該当なし

マニュアルの変更履歴

表 *90* : マニュアルの変更履歴

改訂の詳細	リリース
IPv6 のサポートが導入されました。	2022.04.0
最初の導入。	2021.02.0

機能説明

GTPC および Sx パス管理機能は次をサポートしています。

- エコー要求およびエコー応答メッセージを使用した GTPC パス管理。
- PFCP ハートビート要求とハートビート応答を使用した Sx パス管理。SGW-C と UPF 間のノードレベルのハートビート手順。
- S11 および S5 インターフェイスでの GTPC パス障害の検出。
- Sx インターフェイスでの Sx パス障害の検出。
- パス障害検出機能を設定するためのパス障害検出ポリシーの設定。

GTPC および Sx パスの管理

機能説明

GTPC および Sx パス管理は次をサポートしています。

- ピアの稼働状態をチェックするために、S5 および S11 インターフェイスを介したエコー要求およびエコー応答の交換を使用した GTPC パス管理。
- ピアの稼働状態を確認するために、Sx インターフェイスを介したパケット転送制御プロトコル（PFCP）ハートビート要求とハートビート応答の交換を使用した Sx パス管理。

機能設定

この機能の設定には、次の手順が含まれます。

- エコーパラメータを設定します。詳細については、エコーパラメータの構成（259ページ）を参照してください。

- ハートビートパラメータを設定します。詳細については、ハートビートの設定 （259 ページ） を参照してください。
- ピアの設定を確認します。詳細については、ピア構成の表示 （260 ページ）を参照してください。

エコーパラメータの構成

エコーパラメータを構成するには、次の構成を使用します。

エコー要求の有効化

エコー要求を有効にするには、次の構成を使用します。

```
config
  instance instance-id instance_id
    endpoint endpoint_name
      interface [ s11 | s5e ]
        echo interval interval_value
        echo max-retransmissions max_retransmissions_count
        echo retransmission-timeout retransmission_timeout_count
        end
```

注：

- **interval** *interval_value*：エコー間隔を秒単位で指定します。60 ー 3600 の範囲の整数である必要があります。デフォルト値は 60 秒です。
- **max-retransmissions** *max_retransmissions_count*：GTP エコー要求の最大再試行回数を指定します。0 ～ 15 の範囲の整数で指定する必要があります。デフォルト値は 3 です。
- **retransmission-timeout** *retransmission_timeout_count*：エコー要求の再送信タイムアウト時間を秒単位で指定します。1 ～ 20 の範囲の整数で指定する必要があります。デフォルト値は 5 です。

エコー要求の無効化

エコー要求を無効にするには、次の構成を使用します。

```
config
  instance instance-id instance_id
    endpoint endpoint_name
      replicas replicas_count
      interface interface_name
        no echo
        end
```

ハートビートの設定

ハートビートパラメータを設定するには、次の設定を使用します。

ハートビートの有効化

ハートビートを有効にするには、次の設定を使用します。

```
config
   instance instance-id instance_id
      endpoint pfcp
         interface sxa
            heartbeat
            interval interval
            retransmission-timeout timeout
            max-retransmissions retransmission_count
            end
```

注：

- **interval** *heartbeat_interval*：ハートビート間隔を秒単位で指定します。0 ～ 3600 の範囲の整数で指定する必要があります。無効にするには、0 に設定します。
- **max-retransmissions** *max_retransmissions*：PFCP ハートビート要求の最大再試行回数を指定します。0 ～ 15 の範囲の整数で指定する必要があります。デフォルト値は 4 です。
- **retransmission-timeout** *retransmission_timeout*：ハートビート再送信のタイムアウト期間を秒単位で指定します。1 ～ 20 の範囲の整数で指定する必要があります。デフォルト値は 5 です。

ハートビートの無効化

ハートビートを無効にするには、次の設定を使用します。

```
config
   instance instance-id instance_id
      endpoint pfcp
         interface sxa
            heartbeat
            interval interval
            end
```

注：

- **interval** *heartbeat_interval*：ハートビート間隔を 0 にして、ハートビートを無効にします。

ピア構成の表示

ピアの再起動カウンタを表示するには、次の構成を使用します。

次のコマンドは、ピア構成を表示します。

```
show peers all [ endpoint ] [ local addr ] [ peer addr ]
```

```
show peers all SXA 209.165.201.12:8805 209.165.201.18:8805 POD CONNECTED
ENDPOINT LOCAL ADDRESS PEER ADDRESS DIRECTION INSTANCE TYPE TIME RPC ADDITIONAL DETAILS
```

```
--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
SXA 209.165.201.12:8805 209.165.201.18:8805 Inbound nodemgr-0 Udp 4 hours SGW-U Capacity:
 65535,
LoadMetric: 0,LoadSeqNo: 0,Mode: Online,OverloadMetric: 0,OverloadSeqNo: 0,Priority:
65535

show peers all S11 209.165.201.4:2123 209.165.201.7:2123 LOCAL POD CONNECTED
ADDITIONAL ENDPOINT ADDRESS PEER ADDRESS DIRECTION INSTANCE TYPE TIME RPC DETAILS
--------------------------------------------------------------------------------------
-----------------
S11 209.165.201.4:2123 209.165.201.7:2123 Inbound nodemgr-0 Udp 25 seconds MME Recovery:
 10

show peers all S5E 209.165.201.4:2123 209.165.201.21:2123 LOCAL POD CONNECTED
ADDITIONAL ENDPOINT ADDRESS PEER ADDRESS DIRECTION INSTANCE TYPE TIME RPC DETAILS
--------------------------------------------------------------------------------------
----------------
S5E 209.165.201.4:2123 209.165.201.21:2123 Inbound nodemgr-0 Udp 25 seconds PGW Recovery:
 10

show peers all POD CONNECTED
ENDPOINT LOCAL ADDRESS PEER ADDRESS DIRECTION INSTANCE TYPE TIME RPC ADDITIONAL DETAILS
------------------------------------------------------------------------------------------------------------------------
<none> 209.165.201.29 209.165.201.18:8001 Outbound rest-ep-0 Rest 17 hours UDM <none>
<none> 209.165.201.29 209.165.201.18:8002 Outbound rest-ep-0 Rest 17 hours AMF <none>
<none> 209.165.201.29 209.165.201.18:8003 Outbound rest-ep-0 Rest 17 hours PCF <none>
<none> 209.165.201.29 209.165.201.18:8004 Outbound rest-ep-0 Rest 17 hours CHF <none>
<none> 209.165.201.29 209.165.201.18:9040 Outbound rest-ep-0 Rest 17 hours CHF <none>
S11 209.165.201.4:2123 209.165.201.6:2123 Inbound nodemgr-1 Udp 18 minutes MME Recovery:
 10
S5E 209.165.201.12:2123 209.165.201.24:2123 Inbound nodemgr-1 Udp 5 hours PGW Recovery:
 65535
SXA 209.165.201.12:8805 209.165.201.18:8805 Inbound nodemgr-0 Udp 22 minutes SGW-U
Capacity: 65535,LoadMetric: 0,LoadSeqNo: 0,Mode: Online,OverloadMetric: 0,OverloadSeqNo:
 0,Priority: 65535
```

設定例

エコーを有効にする設定例を次に示します。

```
config
   instance instance-id 1
      endpoint gtp
         interface s11
            echo interval 60
            echo max-retransmissions 5
            echo retransmission-timeout 4
            end
```

エコーを無効にする設定例を次に示します。

```
config
   instance instance-id 1
      endpoint gtp
         replicas 1
         interface s5e
            no echo
            exit
         interface s11
            no echo
            end
```

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

アラート

[Peer Up] と [Peer Down] のアラートを設定するには、『*Cisco Ultra Cloud* サービング ゲートウェイ コントロール プレーンの機能 - メトリックリファレンス』の「重要業績評価指数」の章を参照してください。

バルク統計サポート

Node Manager

次に、エコー送信メッセージおよびエコー再送信メッセージの例を示します。

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_retx",
gtpc_peer_ip="209.165.201.11",instance_id="1",interface_type="S5E",service_name="nodemgr"}
 3
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_tx",
gtpc_peer_ip="209.165.200.230",instance_id="1",interface_type="S11",service_name="nodemgr"}
 2
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_tx",
gtpc_peer_ip="209.165.201.11",instance_id="1",interface_type="S5E",service_name="nodemgr"}
 4
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_tx_initial",
gtpc_peer_ip="209.165.200.230",instance_id="1",interface_type="S11",service_name="nodemgr"}
 2
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_tx_initial",
gtpc_peer_ip="209.165.201.11",instance_id="1",interface_type="S5E",service_name="nodemgr"}
 1
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_res_rx",
gtpc_peer_ip="209.165.200.230",instance_id="1",interface_type="S11",service_name="nodemgr"}
 2
```

GTPC-EP ポッド

次に、受信したエコー要求メッセージと送信されたエコー応答メッセージの例を示します。

```
gtpc_echo_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_rx",
 gtpc_peer_ip="209.165.200.230",instance_id="0",service_name="gtpc-ep"} 1
```

```
gtpc_echo_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_rx",
 gtpc_peer_ip="209.165.200.231",instance_id="0",service_name="gtpc-ep"} 1
```

```
gtpc_echo_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_req_rx",
 gtpc_peer_ip="209.165.201.11",instance_id="0",service_name="gtpc-ep"} 1
```

```
gtpc_echo_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_res_tx",
 gtpc_peer_ip="209.165.200.230",instance_id="0",service_name="gtpc-ep"} 1
```

```
gtpc_echo_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_res_tx",
 gtpc_peer_ip="209.165.200.231",instance_id="0",service_name="gtpc-ep"} 1
```

```
gtpc_echo_msg_stats{app_name="smf",cluster="cn",data_center="cn",gtpc_msg_type="gtpc_echo_res_tx",
 gtpc_peer_ip="209.165.201.11",instance_id="0",service_name="gtpc-ep"} 1
```

プロシージャレベル

次に、ハートビート要求、ハートビート応答、およびハートビート要求の再試行の増分値を確認する方法の例を示します。

```
nodemgr_up_hb_msg_stats{app_name="smf",cluster="Local",current_nodemgr_id="0",data_center="DC",

instance_id="0",interface_type="SXA",primary_nodemgr_id="0",service_name="nodemgr",up_ep_key=
 "209.165.201.1:209.165.201.21",up_msg_type="up_heartbeat_req_retx"} 3
```

```
nodemgr_up_hb_msg_stats{app_name="smf",cluster="Local",current_nodemgr_id="0",data_center="DC",

instance_id="0",interface_type="SXA",primary_nodemgr_id="0",service_name="nodemgr",up_ep_key=
 "209.165.201.1:209.165.201.21",up_msg_type="up_heartbeat_req_tx"} 5
```

```
nodemgr_up_hb_msg_stats{app_name="smf",cluster="Local",current_nodemgr_id="0",data_center="DC",

instance_id="0",interface_type="SXA",primary_nodemgr_id="0",service_name="nodemgr",up_ep_key=
 "209.165.201.1:209.165.201.21",up_msg_type="up_heartbeat_rsp_rx"} 5
```

GTPC パス障害

機能説明

GTPC パス障害では、次の場合に S11 および S5 インターフェイスでピアレベルの GTPC パス障害が検出されます。

- エコー応答に新しい再起動カウンタ値が含まれている場合。
- エコー要求に新しい再起動カウンタ値が含まれている場合。
- エコー応答を受信できない場合。
- Create Session Request または Modify Bearer Request に、新しい再起動カウンタ値が含まれている場合。
- Create Session Response または Modify Bearer Response に、新しい再起動カウンタ値が含まれている場合。

次の場合、異なるパス障害が原因で接続が切断される可能性があります。

- s11_path_failure
- s5e_path-failure
- s11_path_failure_local_purge
- s5e_ path_failure_local_purge
- s5e_recovery
- s11_recovery
- s5e_recovery_local_purge
- s11_recovery_local_purge

機能の仕組み

ここでは、この機能の仕組みを説明します。

GTPC パス障害検出

パス障害は、次の条件で検出されます。

- エコー障害：ピアがエコー要求や再試行に対して応答しない場合、エコー障害が発生します。
- エコー応答または制御メッセージでのカウンタの再起動：GTPC エンティティは、エコー応答内、またはピア GTPC メッセージからリカバリ IE を受信すると、受信した再起動カ

ウンタ値を、そのピアエンティティに以前保存されていた再起動カウンタ値と比較し、次の操作を実行します。

- 以前保存されていた値が使用できない場合、ピアの受信した再起動カウンタ値を保存します。
- max-remote-rc-change パラメータが設定されていない場合、GTPC が再起動カウンタの変更を検出します。
- max-remote-rc-change が設定されている場合、再起動カウンタのロールオーバーを考慮して、再起動カウンタ値の差を計算します。新しい再起動カウンタと古い再起動カウンタの差が、max-remote-rc-change の値よりも小さい場合、パス障害が検出されます。

（注） max-remote-rc-change の詳細については、パス障害検出のカスタマイズ （269 ページ）を参照してください。

パスの障害の処理

パスの障害を検出すると、ネットワークノードは運用および保守システムを介して障害を通知し、次の処理を実行します。

- PDN 接続（EPS ベアラーコンテキスト）またはピア IP アドレスを持つ関連する PDP コンテキストを削除します。
- 選択したインターフェイスに対して次のアクションを指定します。
  - [Local Purge]：cnSGW-C は、ピアに通知せずに、影響を受けるベアラー（またはデフォルトのベアラーがパスの障害を受信した場合は PDN）をクリアします。このアクションは、すべてのインターフェイスに対してデフォルトです。

（注） cnSGW-C は、パス障害の検出時に Sx セッション削除要求を UPF に送信してセッションをクリアします。

  - [Signal-Peer]：cnSGW-C はピア MME と P-GW に向けて制御信号を送信します。

    信号送信時：

    - PDN の削除の場合、SGW は PGW にセッション削除要求メッセージを送信し、MME にベアラー削除要求（LBI を使用）メッセージを送信します。
    - SGW は、S11 または S5 インターフェイス上で削除要求を送信し、ピアに通知します。

（注） ピアが削除されると、エコー要求の交換が停止します。

機能設定

この機能の設定には、次の手順が含まれます。

- パス障害検出で実行する必要があるアクションを構成します。詳細については、パス障害検出時のアクションの構成 （266 ページ） を参照してください。
- ピアノードを更新するための通知を構成します。詳細については、ピアノードを更新する通知の設定 （266 ページ） を参照してください。
- 設定を確認します。詳細については、設定例 （266 ページ）を参照してください。

パス障害検出時のアクションの構成

パス障害検出のアクションを構成するには、次の構成を使用します。

```
config
  profile sgw sgw_name
   path-failure [ s11 | s5e ] [ local-purge | signal-peer ]
   end
```

ピアノードを更新する通知の設定

cnSGW-C が再起動するたびに、再起動カウンタを更新する必要があります。この機能を実装する場合は、Ops Center で Kubernetes の use-volume-claims パラメータ値が true に設定されていることを確認します。

この設定により、CLI システムモードがシャットダウンされ、システムモードが実行中の状態で cnSGW-C が再起動すると、再起動カウンタが更新されます。

設定例

次に、パス障害検出の設定例を示します。

```
config
profile sgw sgw1
 path-failure s11 local-purge
 path-failure s5e local-purge
exit
```

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計サポート

次に、GTPC パス障害の例を示します。

```
nodemgr_gtpc_pathfail_reasons{app_name="smf",cluster="cn",
data_center="cn",instance_id="1",pathfail_reason="pathfail_no_echo_rcv",
service_name="nodemgr"} 2
```

```
nodemgr_gtpc_pathfail_reasons{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pathfail_reason="pathfail_echo_res_rc_change",
service_name="nodemgr"} 1
```

Sx パス障害

機能説明

Sx パス障害では、次の場合に Sx インターフェイスでパス障害が検出されます。

- ハートビート要求に高い値のリカバリタイムスタンプが含まれている。
- ハートビート応答に高い値のリカバリタイムスタンプが含まれている。
- ハートビート応答を受信していない。
- Sx 関連付け要求を受信した。
- cnSGW-C は、ピアを解放するための Sx 関連付け更新要求を受信します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

ハートビート要求

cnSGW-C または UPF は、ピアノードへのパスでハートビート要求を送信して、ノードが動作しているかどうかを調べます。ハートビート要求メッセージは、PFCP 制御アソシエーションが確立されている各ピアに送信されます。 cnSGW-C または UPF は、ハートビート要求を受信する準備が整っており、ハートビート応答で応答します。ハートビート要求は、ピアとの新しいセッションが確立されるとピアで開始し、最後のセッションがピアから解放されると停止します。

cnSGW-C と UPF は、構成された間隔に基づいてハートビート要求を送信します。ピアが応答しない場合、メッセージは、再試行間隔内で構成された回数だけ再試行されます。応答を受信すると、対応するピアに関連付けられたコールに対して定義されたアクションが実行されます。

リカバリタイムスタンプは、ピアノードの開始時刻を含む IE です。ハートビート要求には、ピアに送信される自己回復タイムスタンプ値が含まれています。

（注） ハートビート要求は、ピアが削除された場合にのみ停止します。

ハートビート応答

ハートビート応答メッセージは、受信したハートビート要求への応答として送信されます。

リカバリタイムスタンプは、ノードの開始時刻を含むIEです。ハートビート応答には、ピアのリカバリタイムスタンプ値が含まれています。

Sx パス障害の検出

Sx パス障害は、次の条件で検出されます。

- **ハートビート障害**：ピアが、送信されたハートビートおよび再試行に応答しない場合。
- **ハートビートのリカバリタイムスタンプの変更**：ハートビート応答に新しいリカバリタイムスタンプ値がある場合は、以前に受信した値になります。受信したリカバリタイムスタンプの値が以前に受信した値よりも小さい場合、パス障害は検出されません。
- **Sx 関連付けメッセージ**：ピアから Sx 関連付けメッセージを再受信した場合。この場合、すべてのコールがクリアされ、通知が eGTP ピアに送信されます。
- **Sx 関連付けリリースメッセージ**：Sx 関連付けリリースメッセージを受信した場合。この場合、すべてのコールがクリアされ、通知が eGTP ピアに送信されます。

パスの障害の処理

受信したリカバリタイムスタンプの値が以前に受信した値より大きい場合、ピアの再起動が検出されます。タイムスタンプが以前に受信した値よりも小さい場合、その値は無視され、ピアの再起動は検出されません。

ピアのパス障害を示すピアの再起動が検出されると、そのピアに接続されているすべてのコールがクリアされます。このようなコールで使用される切断理由は、Sx パス障害です。

Sx パス障害が検出されると、Sx 関連付けも削除されます。

ハートビート処理

PFCP エンティティは、（不明なピアからの場合でも）ハートビート要求メッセージを受信するたびに、ハートビート応答メッセージで応答します。

No response to peer エラーが原因のパス障害が検出されると、関連付けが再確立されるまで、そのピアにハートビート要求は送信されません。コールは、パス障害検出ポリシーの設定に基づいてクリアされます。

（注） Sx の関連付けが削除された後、Sx パス障害が検出されると、ハートビートが停止します。

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

次に、ハートビート要求、ハートビート応答、およびハートビート要求の再試行で増分されるプロシージャレベルの統計情報の例を示します。

```
nodemgr_up_hb_msg_stats{app_name="smf",cluster="Local",current_nodemgr_id="0",
data_center="DC",instance_id="0",interface_type="SXA",primary_nodemgr_id="0",service_name=
"nodemgr",up_ep_key="209.165.201.5:209.165.201.28",up_msg_type="up_heartbeat_req_retx"}
 3
```

```
nodemgr_up_hb_msg_stats{app_name="smf",cluster="Local",current_nodemgr_id="0",
data_center="DC",instance_id="0",interface_type="SXA",primary_nodemgr_id="0",service_name=
"nodemgr",up_ep_key="209.165.201.5:209.165.201.28",up_msg_type="up_heartbeat_req_tx"}
 5
```

```
nodemgr_up_hb_msg_stats{app_name="smf",cluster="Local",current_nodemgr_id="0",
data_center="DC",instance_id="0",interface_type="SXA",primary_nodemgr_id="0",service_name=
"nodemgr",up_ep_key="209.165.201.5:209.165.201.28",up_msg_type="up_heartbeat_rsp_rx"}
 5
```

パス障害検出のカスタマイズ

機能説明

cnSGW-C により、パス障害検出ポリシーを設定できます。デフォルトでは、パス障害検出ポリシーは有効になっています。

- **GTPC パス障害検出のカスタマイズ**：GTPC パス障害は、次の場合に検出されます。
  - エコー要求の再試行回数が上限に達した。
  - エコー要求または応答の再起動カウンタが変更された。

- 制御メッセージ応答の再起動カウンタが変更された。
- 新しい再起動カウンタと古い再起動カウンタの絶対差が、max-remote-rc-change で設定された値よりも小さい場合。

（注） GTPCパス障害検出のカスタマイズにより、ユーザーは最大リモート再起動カウンタ（max-remote-rc-change）変更機能を使用して、ピアの誤った再起動を無視できます。

- **Sx パス障害検出のカスタマイズ**：PFCP パス障害は、次の場合に検出されます。
  - ハートビート要求の再試行回数が上限に達した。
  - ハートビート要求または応答のリカバリタイムスタンプが変更された。

機能設定

この機能の設定には、次の手順が含まれます。

- GTPC パス失敗のカスタマイズを構成します。詳細については、GTPC パス障害のカスタマイズの構成 （270 ページ） を参照してください。
- Sx パス障害のカスタマイズを構成します。詳細については、Sx パス障害のカスタマイズの設定 （270 ページ）を参照してください。

Sx パス障害のカスタマイズの設定

Sx パス障害のカスタマイズを設定するには、次の設定を使用します。

```
config
 policy sx-path-failure-detection policy
  ignore heartbeat-retry-failure
  ignore heartbeat-recovery-timestamp-change
exit
 instance instance-id instance_id
  endpoint pfcp
    replicas replica_count
    sx-path-failure sx-detection-policy policy
    interface sxa
    sx-path-failure sx-detection-policy policy
    end
```

GTPC パス障害のカスタマイズの構成

GTPC パス障害のカスタマイズを構成するには、次の構成を使用します。

```
config
  policy path-failure-detection policy_name
    max-remote-rc-change maximum_remote
    ignore echo-rc-change
    ignore control-rc-change
    ignore echo-failure
  exit
 exit
 instance instance-id instance_id
  endpoint gtp
    replicas replica_count
    vip-ip ipv4_address vip-port ipv4_port_number
    vip-ipv6 ipv6_address vip-ipv6-port ipv6_port_number
    dual-stack-transport { true | false }
    path-failure detection-policy policy
    interface [ s11 | s5e ]
    end
```

注：

- GTPC パス障害検出ポリシーがインターフェイスレベルで構成されていない場合、エンドポイントレベルのパス障害検出ポリシーを適用できます。
- max-remote-rc-change 構成は、S11 または S5 がピアの再起動を検出するまでのカウンタの変更を指定します。新しい再起動カウンタと古い再起動カウンタの絶対差が構成された値よりも小さい場合にのみ、ピアの再起動が検出されますたとえば、max-remote-rc-change が 10 で、現在のピアの再起動カウンタが 251 の場合は、新しい再起動カウンタが 252 ～ 255 または 0 ～ 5 の場合にのみ、eGTP はピアの再起動を検出します。同様に、保存された再起動カウンタが 1 の場合は、新しい再起動カウンタが 2 ～ 11 の場合にのみ、eGTP はピアの再起動を検出します。
- 有効な設定値は、1 ～ 255 です。推奨設定値は 32 です。

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

GTPC パス障害

エコー要求/応答メッセージや、コントロール要求/応答メッセージにおける再起動カウンタの変化によってパス障害が検出された回数を示す統計情報を保持します。

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="Local",data_center="DC",
gtpc_msg_type="gtpc_false_peer_restart_cfg_ctrl_rc_change",gtpc_peer_ip=
"209.165.201.17",instance_id="0",interface_type="S11",service_name="nodemgr"}
 1
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="Local",data_center="DC",
gtpc_msg_type="gtpc_false_peer_restart_cfg_echo_rc_change",gtpc_peer_ip=
"209.165.201.17",instance_id="0",interface_type="S11",service_name="nodemgr"}
 1
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="Local",data_center="DC",
gtpc_msg_type="gtpc_false_peer_restart_cfg_echo_rc_change",gtpc_peer_ip=
"209.165.201.27",instance_id="0",interface_type="S5E",service_name="nodemgr"}
 1
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="Local",data_center="DC",
gtpc_msg_type="gtpc_ignore_echo_timeout",gtpc_peer_ip="209.165.201.27",
instance_id="0",interface_type="S5E",service_name="nodemgr"} 1
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="Local",data_center="DC",
gtpc_msg_type="gtpc_false_peer_restart_ignore_echo_rc_cfg",gtpc_peer_ip=
"209.165.201.27",instance_id="0",interface_type="S5E",service_name="nodemgr"}
 1
```

```
nodemgr_gtpc_msg_stats{app_name="smf",cluster="Local",data_center="DC",
gtpc_msg_type="gtpc_false_peer_restart_ignore_ctrl_rc_cfg",gtpc_peer_ip=
"209.165.201.27",instance_id="0",interface_type="S5E",service_name="nodemgr"}
 1
```

表 ***91 : GTPC*** パス障害の統計情報の説明

統計	説明
gtpc_false_peer_restart_cfg_echo_rc_change	エコー再起動カウンタの変更が設定済みの max-remote-rc-change の範囲内にないために無視された GTPC パス障害の数。
gtpc_false_peer_restart_ignore_echo_rc_cfg	エコー再起動カウンタが変更されたために無視された GTPC パス障害の数。
gtpc_false_peer_restart_cfg_ctrl_rc_change	制御メッセージ再起動カウンタの変更が設定済みの max-remote-rc-change の範囲内にないために無視された GTPC パス障害の数。
gtpc_false_peer_restart_ignore_ctrl_rc_cfg	制御メッセージ再起動カウンタが変更されたために無視された GTPC パス障害の数。
gtpc_ignore_echo_timeout	エコー要求のタイムアウトによって無視された GTPC パス障害の数。

Sx パス障害

次のメッセージで、リカバリタイムスタンプの変化によってパス障害が検出された回数を示す統計情報を保持します。

```
nodemgr_up_pathfail_reasons{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",service_name="nodemgr",up_pathfail_reason="up_pathfail_ignored_hb_retry"}
 1
```

```
nodemgr_up_pathfail_reasons{app_name="smf",cluster="cn",data_center="cn",
instance_id="1",service_name="nodemgr",up_pathfail_reason="up_pathfail_ignored_hb_rt_change"}
 1
```

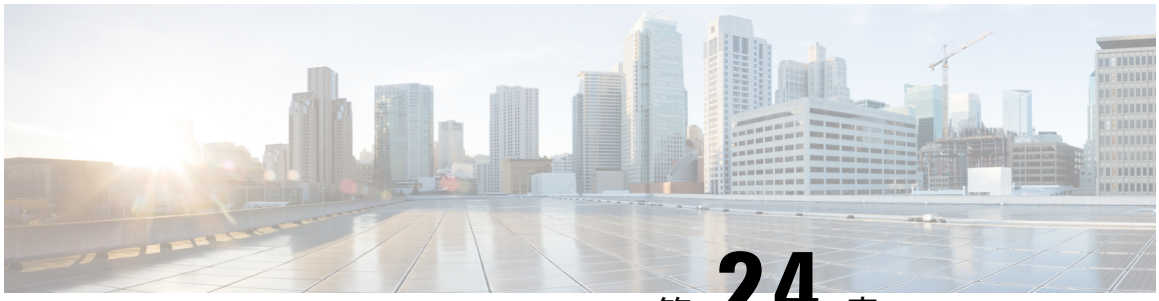
```
nodemgr_up_pathfail_reasons{app_name="smf",cluster="cn",data_center="cn",
instance_id="1",service_name="nodemgr",up_pathfail_reason="up_pathfail_reason_association_release"}
 1
```

```
nodemgr_up_pathfail_reasons{app_name="smf",cluster="cn",data_center="cn",
instance_id="1",service_name="nodemgr",up_pathfail_reason="up_pathfail_reason_hb_retry"}
 8
```

```
nodemgr_up_pathfail_reasons{app_name="smf",cluster="cn",data_center="cn",
instance_id="1",service_name="nodemgr",up_pathfail_reason="up_pathfail_reason_hb_rt_change"}
 1
```

表 *92 : Sx* パス障害の統計情報の説明

統計	説明
up_pathfail_ignored_hb_retry	ハートビート要求のタイムアウトによって無視された Sx パス障害の数。
up_pathfail_reason_hb_retry	ハートビート要求のタイムアウトによって検出された Sx パス障害の数。
up_pathfail_ignored_hb_rt_change	ハートビート要求のリカバリタイムスタンプの変更によって無視された Sx パス障害の数。
up_pathfail_reason_hb_rt_change	ハートビート要求のリカバリタイムスタンプの変更によって検出された Sx パス障害の数。
up_pathfail_reason_association_release	Sx の関連付けの解除が原因で検出された Sx パス障害の数。



第 24 章

GTPU エラーの表示

- 機能の概要と変更履歴 （275 ページ）
- 機能説明 （276 ページ）
- 機能の仕組み （276 ページ）
- 機能設定 （295 ページ）
- OAM のサポート （296 ページ）

機能の概要と変更履歴

要約データ

表 *93* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *94* : マニュアルの変更履歴

改訂の詳細	リリース
信号ピアのサポートが追加されました（エラー表示と構成は信号ピア）	2021.02.3
最初の導入。	2021.02.1

機能説明

cnSGW-C は、セッションレポート要求で UPF によって報告された GTPU エラーをサポートしています。UPF は、PFCP セッションレポート要求メッセージで CP（cnSGW-C）にさまざまな GTPU エラーを報告します。

cnSGW-C は、次のレポート要求をサポートしています。

- エラー表示のサポート（ERIR）
- グレースフル終了（GTER）
- セッションの再配置（SRIR）

機能の仕組み

ここでは、この機能の仕組みを説明します。

エラー表示のサポート

cnSGW-C は、UPF から PFCP セッションレポート要求とともにエラー表示を受信すると、PFCP セッションレポート応答で応答し、構成に従って動作します。

cnSGW-C の場合、シグナリングは設定に基づいています。

- S1U：ローカルパージまたは page-ue
- S5U：ローカルパージまたはピアシグナリング

表 *95* : エラー表示サポート（*ERIR*）レポートタイプ

インターフェイス	設定	TEID	Action
S1U	ローカルパージ	デフォルト	SxSessionDeleteRequest を送信して UPF をクリーンアップする ローカルでパージ
		Dedicated	SxModReq の送信（トラフィックエンドポイントの削除） ローカルパージ
		IDFT	SxMod の送信（IDFT トラフィックエンドポイントの削除）：非同期 ベアラーをローカルでパージ
	Page-UE	デフォルト/専用	UE をアイドル状態に移行 Sx_Modification の送信 _Request (Set FAR Action=BUFFER)
S5U	ローカルパージ	デフォルト	SxDeleteSession の送信 ローカルでパージ
		Dedicated	SxMod の送信（トラフィックエンドポイントの削除） ベアラーをローカルでパージ
	シグナルピア	デフォルト	SxMod（ドロップ）DBR/DSR、SxDelete の送信
		Dedicated	DBR/DBC（非同期）、SxMod（トラフィックエンドポイントの削除）の送信

local-purge として s1u を使用するデフォルト ベアラー コール フロー

ここでは、local-purge として s1u を使用するデフォルト ベアラー コール フローについて説明します。

図 *44 : local-purge* として *s1u* を使用するデフォルト ベアラー コール フロー

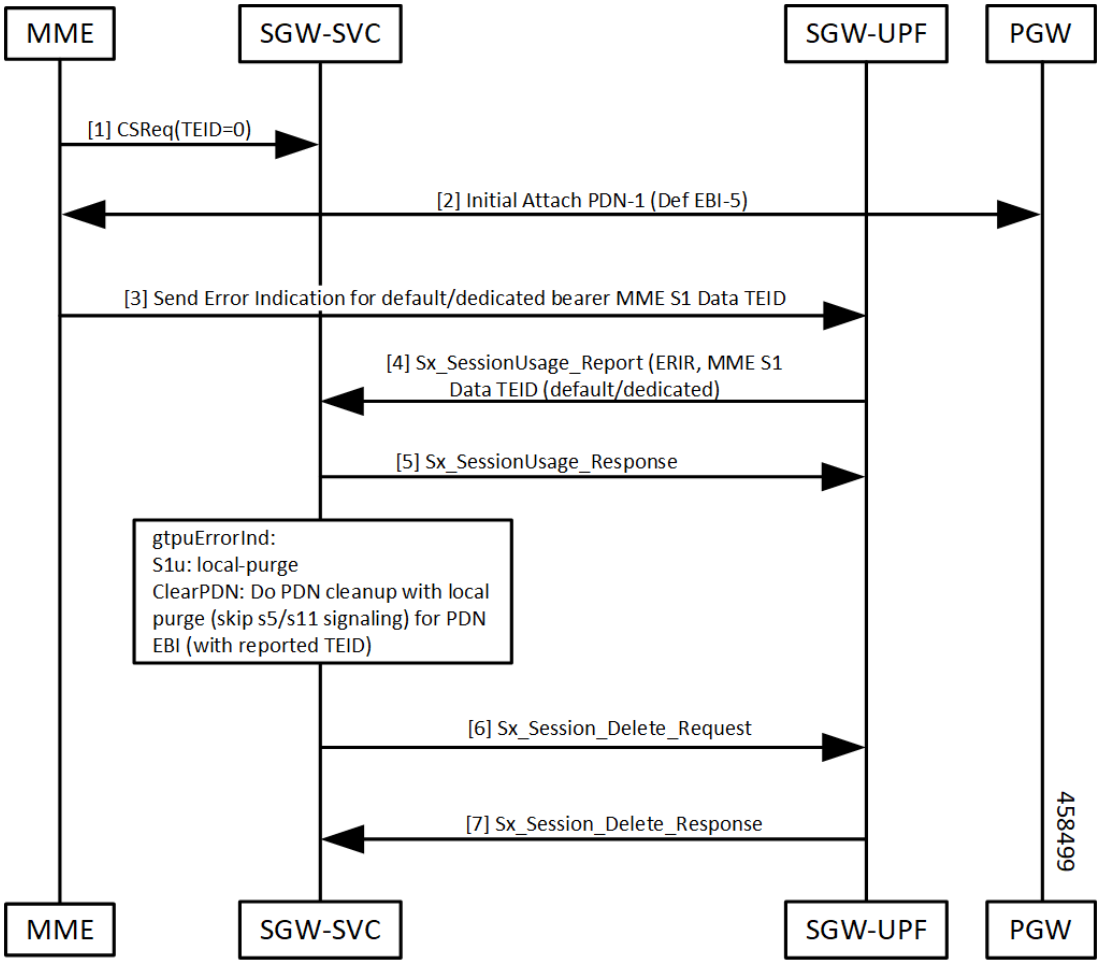


表 *96 : local-purge* として *s1u* を使用するデフォルト ベアラー コール フローの説明

ステップ	説明
1、2	初期接続が完了しました。
3、4、5	• UPF が、レポートタイプを ERIR および s1u TEID として Sx_Session_Report_Request を送信します。 • cnSGW-C が Sx_Session_Report_Response で応答します。

ステップ	説明
6、7	cnSGW-C は Sx_Session_Report_Request を処理します。 gtpuErrorInd： s1u：local-purge デフォルトベアラーの TEID を受信した場合は、内部トランザクション（T2）がクリーンアップベアラーに送信されます（ピアシグナリングなし）。 • Sx_Session_Deletion_Request を送信します。 • UPF は Sx_Session_Deletion_Response で応答します。

s1u を local-purge コールフローとして使用する専用ベアラー

このセクションでは、s1u を local-purge コールフローとして使用する専用ベアラーについて説明します。

図 *45 : s1u* を *local-purge* コールフローとして使用する専用ベアラー

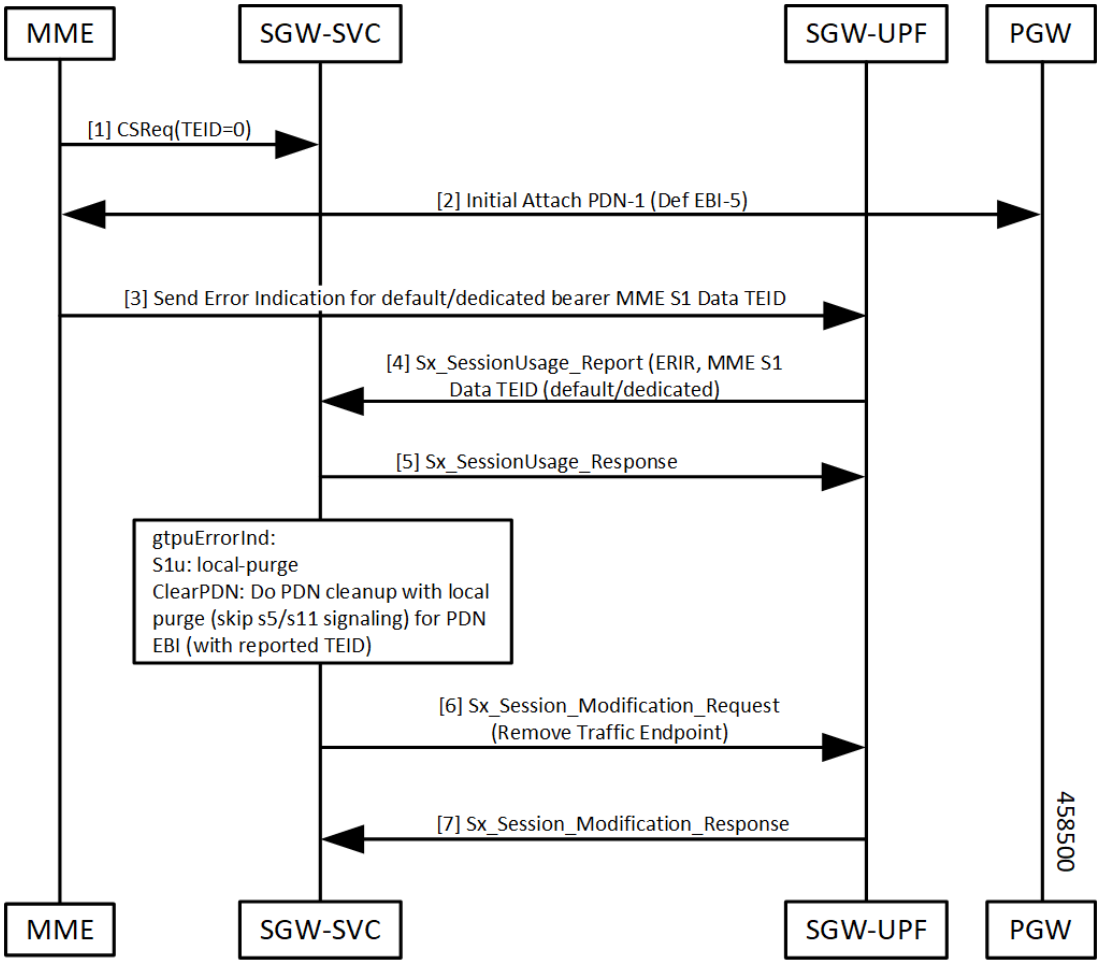


表 *97 : s1u* を *local-purge* コールフローの説明として使用する専用ベアラ

ステップ	説明
1、2	初期接続が完了しました。
3、4、5	• UPF は、レポートタイプを ERIR および s1u TEID として Sx_Session_Report_Request を送信します。 • cnSGW-C が Sx_Session_Report_Response で応答します。

ステップ	説明
6、7	cnSGW-C は Sx_Session_Report_Request を処理します。 gtpuErrorInd： s1u：local-purge TEID を専用ベアラー用に受信した場合は、内部トランザクション（T2）をクリーンアップベアラー（ピアシグナリングなし）に送信します。 • Sx_Session_Modification_Request（Remove Traffic Endpoint）を送信します。 • UPF が Sx_Session_Modification_Response で応答します。

s1u を local-purge コールフローとして使用する専用ベアラー（IDFT）

このセクションでは、s1u を local-purge コールフローとして使用する専用ベアラー（IDFT）について説明します。

図 *46 : s1u* を *local-purge* コールフローとして使用する専用ベアラー（*IDFT*）

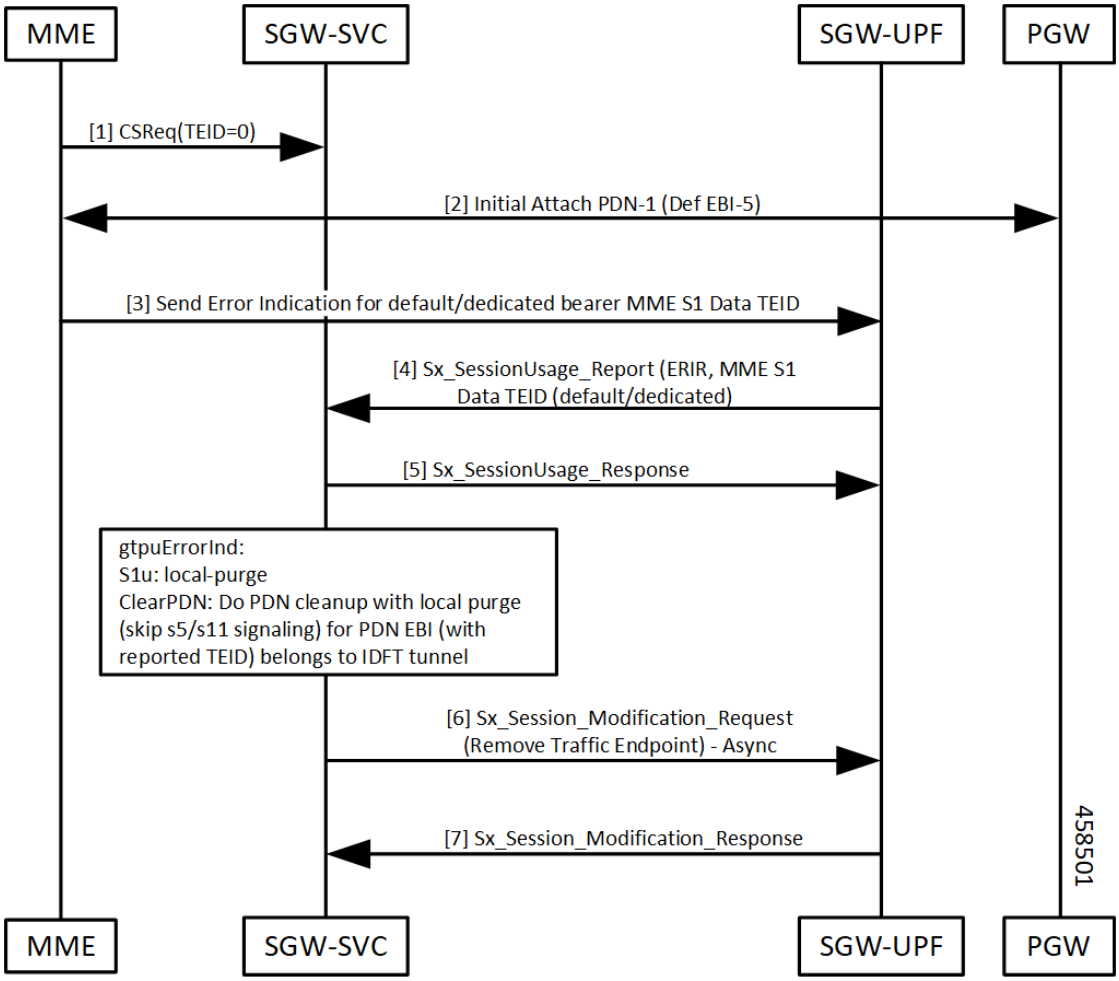


表 *98 : s1u* を *local-purge* コールフローとして使用する専用ベアラー（*IDFT*）の説明

ステップ	説明
1、2	初期接続が完了しました。
3、4、5	• UPF が、レポートタイプを ERIR および s1u TEID として Sx_Session_Report_Request を送信します。 • cnSGW-C が Sx_Session_Report_Response で応答します。

ステップ	説明
6、7	cnSGW-C は Sx_Session_Report_Request を処理します。 gtpuErrorInd： s1u：local-purge TEID を専用ベアラー（IDFT）用に受信した場合は、内部トランザクション（T2）をクリーンアップベアラー（ピアシグナリングなし）に送信します。 • Sx_Session_Modification_Request（トラフィックエンドポイントの削除）を送信します。 • UPF が Sx_Session_Modification_Response で応答します。

s1u を page-ue として使用するデフォルト/専用ベアラーコールフロー

このセクションでは、s1u を page-ue として使用するデフォルト/専用ベアラーコールフローについて説明します。

図 *47 : s1u* を *page-ue* として使用するデフォルト/専用ベアラーコールフロー

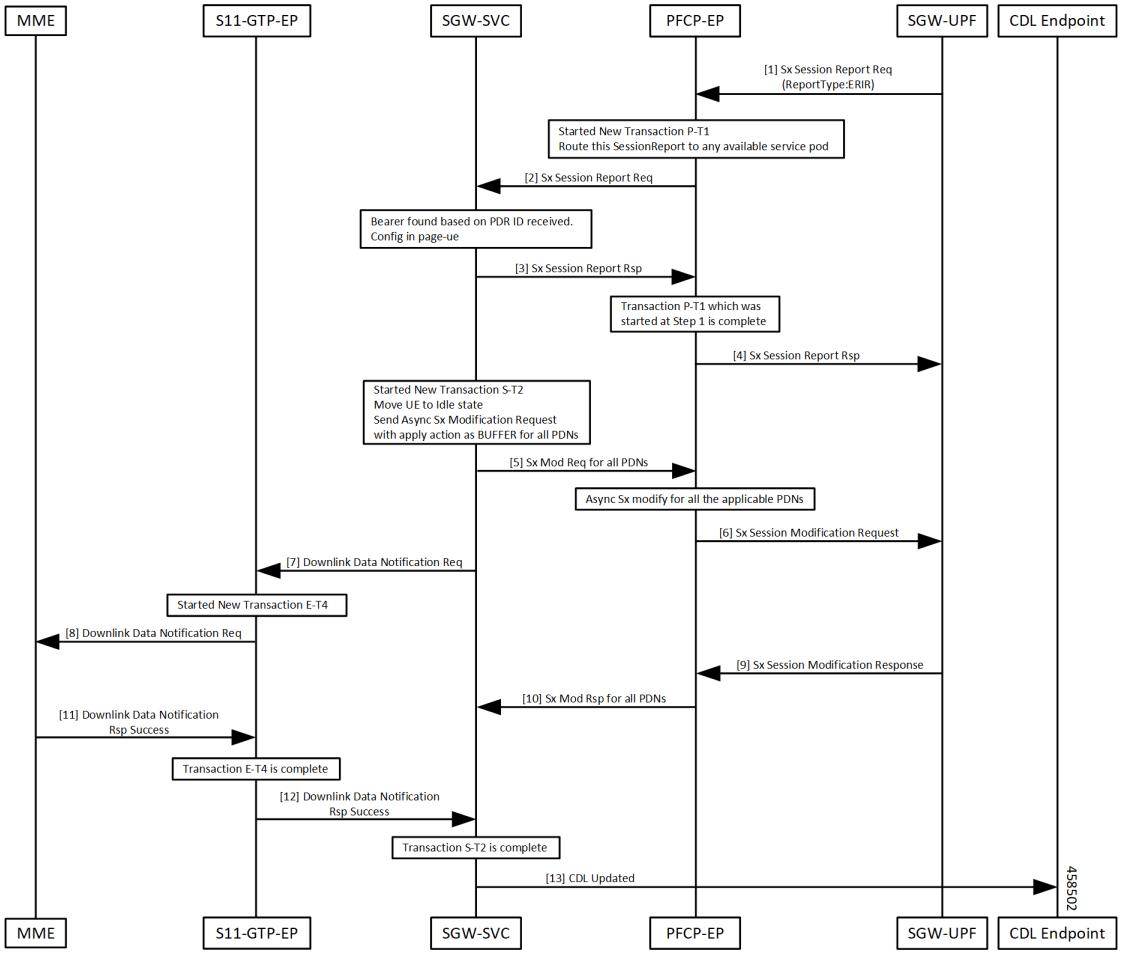


表 *99 : s1u* を *page-ue* として使用するデフォルト/専用ベアラーコールフローの説明

ステップ	説明
1、2	SGW-UP が、タイプが ERIR の Sx セッションレポート要求を SGW サービスポッドに送信します。
3、4	SGW サービスポッドが SGW-UPF に Sx セッションレポート応答を送信します。
5、6	PFCP-EP が SGW サービスポッドにすべての PDN の Sx 変更要求を送信します。 PFCP-EP が SGW-UP に Sx セッション変更要求を送信します。
7、8	SGW サービスポッドが S11-GTP-EP にダウンリンクデータ通知要求を送信します。 S11-GTP-EP が MME にダウンリンクデータ通知を転送します。

ステップ	説明
9、10	SGW-UP が SGW サービスポッドに Sx セッション変更応答を送信します。
11、12、13	MME がダウンリンクデータ通知応答の成功を S11-GTP-EP に送信します。 S11-GTP-EP がダウンリンクデータ通知応答の成功を SGW サービスポッドに転送します。 S-T2 トランザクションが完了すると、SGW サービスポッドが CDL エンドポイントに CDL の更新を送信します。

local-purge/signal-peer として s5u を使用するデフォルト ベアラー コール フロー

ここでは、local-purge/signal-peer として s5u を使用するデフォルト ベアラー コール フローについて説明します。

図 *48 : local-purge/signal-peer* として *s5u* を使用するデフォルト ベアラー コール フロー

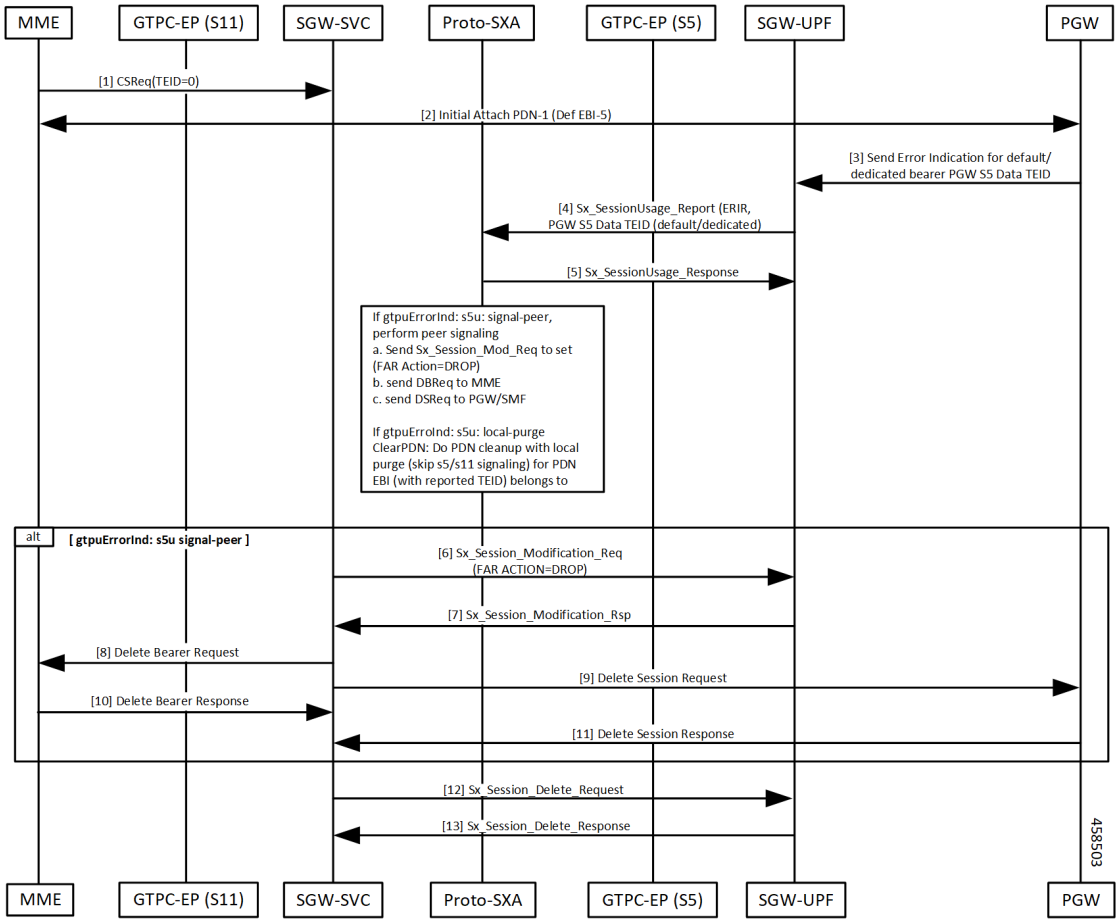


表 *100 : local-purge/signal-peer* として *s5u* を使用するデフォルト ベアラー コール フローの説明

ステップ	説明
1、2	初期接続が完了しました。
3、4、5	UPF で GTPU エラーが検出されました。 • Sx_Session_Report_Request が cnSGW-C に送信されました。 • cnSGW-C は Sx_Session_Resport_Response で応答します。
6 ～ 11	cnSGW-C がセッションレポート要求（ERIR）を処理します。 デフォルトベアラーの TEID を受信した場合、内部トランザクションを送信して PDN をクリーンアップします（動作は CLI の構成によって異なります）。 CLI：sgw-profile config gtpuErrorInd の場合： s5u：signal-peer • Sx_Session_Report_Request を UPF に送信し、(FAR ACTION=DROP) を設定します。 • ベアラー削除要求を MME に送信します。 • セッション削除要求を PGW に送信します。
12	Sx_Session_Delete_Request を UPF に送信します。
13	UPF が Sx_Session_Delete_Response で応答します。

local-purge/signal-peer として s5u を使用する専用ベアラーコールフロー

ここでは、local-purge/signal-peer として s5u を使用する専用ベアラコールフローについて説明します。

図 *49 : local-purge/signal-peer* として *s5u* を使用する専用ベアラーコールフロー

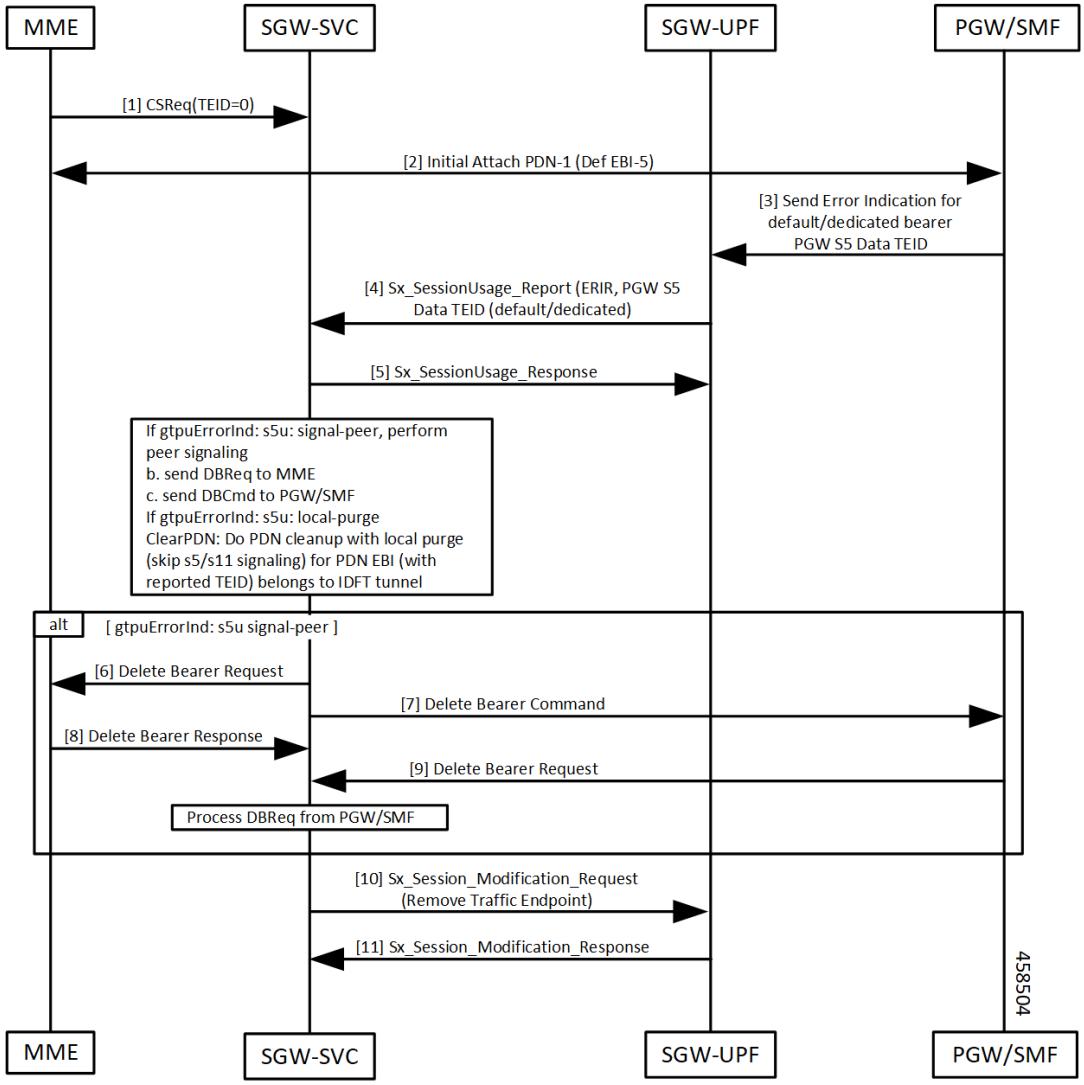


表 *101 : local-purge/signal-peer* として *s5u* を使用する専用ベアラーコールフローの説明

ステップ	説明
1、2	初期接続が完了しました。
3、4、5	UPF で GTPU エラーが検出されました。 • Sx_Session_Report_Request が cnSGW-C に送信されました。 • cnSGW-C が Sx_Session_Report_Response で応答します。

ステップ	説明
6、7、8、9	cnSGW-C が Session Report Request（ERIR）を処理します。専用ベアラー（s5u）の TEID を受信した場合は、ベアラーをクリーンアップするために、内部トランザクションを送信します。 CLI：sgw-profile config gtpuErrorInd の場合： s5u：signal-peer • Delete Bearer Req を MME に送信します。 • Delete Bearer Command を PGW に送信します。
10	Sx_Session_Modification_Request（Remove Traffic Endpoint）を UPF に送信します。
11	UPF が Sx_Session_Modification_Response で応答します。

グレースフル終了

UPF は SR/ICSR のリカバリ中に PDU セッションをリカバリできない場合、タイプをグレースフル終了レポート（GTER）として、PFCP Session Report Request を cnSGW に送信します。

UPF はセッションリカバリ中にセッションをロードできない場合、報告されたセッションに関するすべてのインターフェイスをクリアすることを示す GTER を送信します。

グレースフル終了のコールフロー

ここでは、グレースフル終了フローについて説明します。

図 *50 :* グレースフル終了のコールフロー

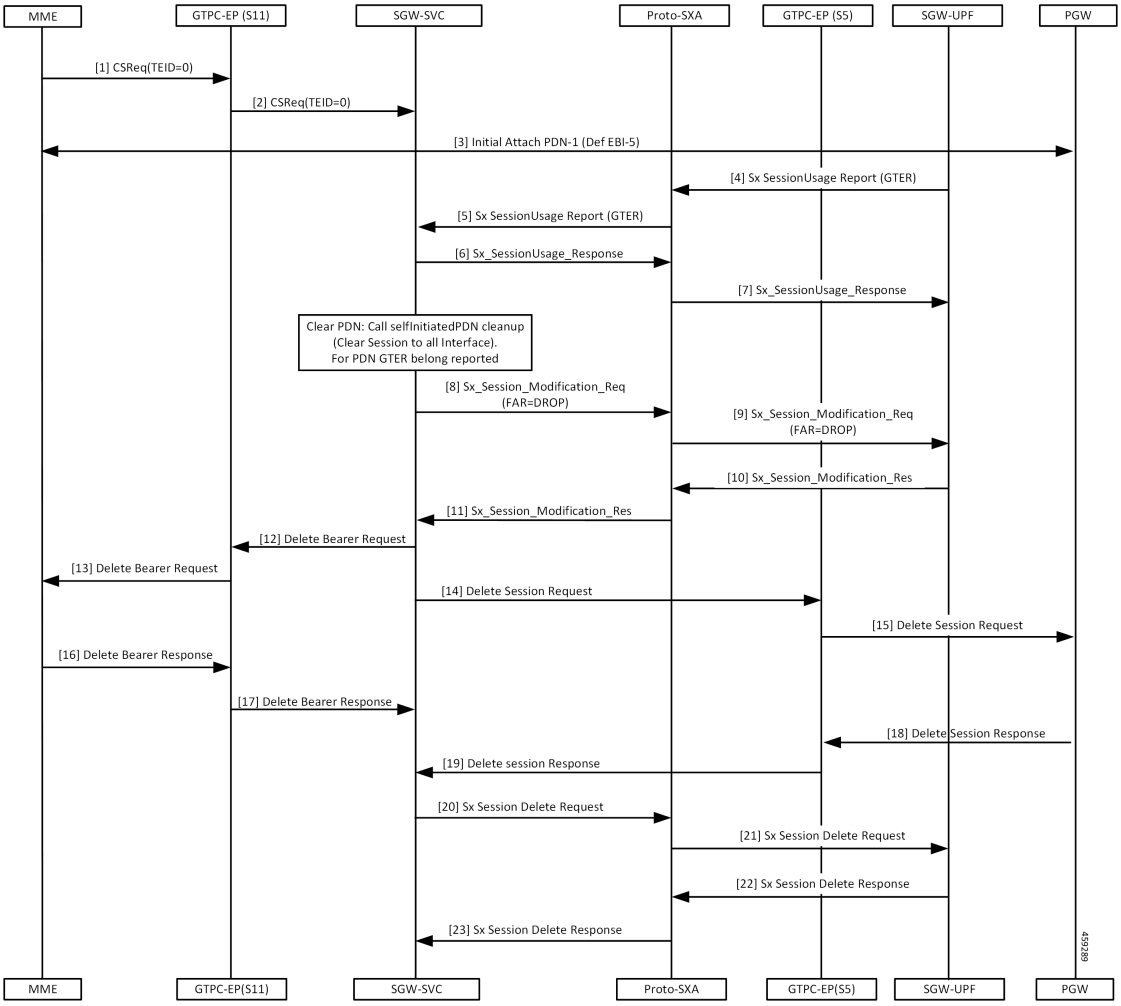


表 *102 :* グレースフル終了のコールフローの説明

ステップ	説明
1、2、3	初期接続が完了しました。
4-7	• UPF が、レポートタイプを GTER および TEID として Sx_Session_Report_Request を送信します。 • cnSGW-C が Sx_Session_Report_Response で応答します。

ステップ	説明
8-11	cnSGW-C が Sx_Session_Report_Request を処理し、内部トランザクション（T2）を送信して PDN をクリーンアップします。 • cnSGW-C が Sx_Session_Modification_Request を送信して、FAR Action=Drop を設定します。 • UPF が Sx_Session_Modification_Response で応答します。
12-19	• ベアラー削除要求を MME に送信します。 • セッション削除要求を PGW に送信します。
20-23	• Sx_Session_Delete_Request を送信します。 • UPF が Sx_Session_Delete_Response で応答します。

セッションの再配置

ピアが同じ GTP-U TEID を割り当てる場合は、セッションの再配置（SRIR）が必要です。

UPF は、同じ TEID を持つ古いセッションを削除するように示す SRIR レポートを送信します。cnSGW-C は、SRIR 要求処理に GTPU パス障害構成を使用します。

表 *103* : セッションの再配置

インターフェイス	設定	TEID	Action
S1U/S5U	ローカルパージ	デフォルト	SxDeleteSession の送信 ローカルでパージ
		Dedicated	SxMod の送信（トラフィックエンドポイントの削除） ベアラーをローカルでパージ
	シグナルピア	デフォルト	SxModの送信（ドロップ） DBR/DSR SxDelete
		Dedicated	DBR/DBC の送信（非同期） SxMod（トラフィックエンドポイントの削除）
IDFT	該当なし	該当なし	SxMod の送信（IDFT トラフィックエンドポイントの削除）：非同期 ベアラーをローカルでパージ

デフォルトベアラーのセッション置換コールフロー

ここでは、デフォルトベアラーのセッション置換（SRIR）コールフローについて説明します。

図 *51* : デフォルトベアラーのセッション置換コールフロー

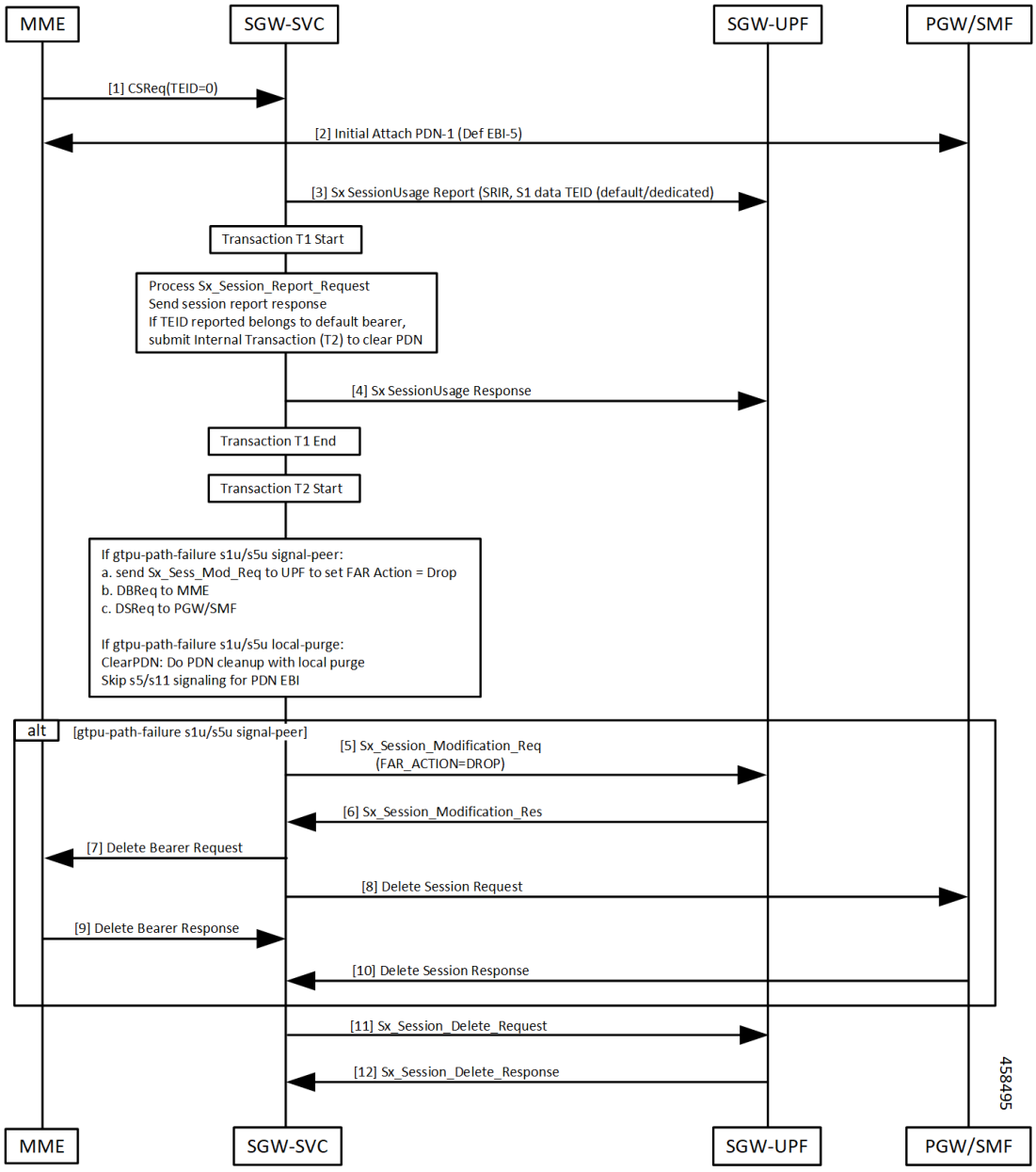


表 *104* : デフォルトベアラーのセッション置換コールフローの説明

ステップ	説明
1、2	初期接続が完了しました。

ステップ	説明
3、4	• UPF がレポートタイプが SRIR および TEID の Sx_Session_Report_Request を送信します。 • cnSGW-C が Sx_Session_Report_Response で応答します。
5 ～ 10	cnSGW-C が Sx_Session_Report_Request を処理し、誤ったセッション置換でピアシグナリングまたはローカル消去のために GTPU パス障害 CLI を使用します。 デフォルトベアラーの TEID を受信すると、内部トランザクション（T2）を送信して PDN をクリーンアップします。 CLI の場合、`gtpu-path-failure s1u/s5u signal-peer` • Sx_Session_Report_Request を UPF に送信し、FAR ACTION=DROP を設定します。 • Delete Bearer Req を MME に送信します。 • セッション削除要求を PGW に送信します。
11、12	• Sx_Session_Delete_Request を送信します。 • UPF が Sx_Session_Delete_Response で応答します。

専用ベアラーのセッション置換コールフロー

ここでは、専用ベアラーのセッション置換（SRIR）コールフローについて説明します。

図 *52 :* 専用ベアラーのセッション置換コールフロー

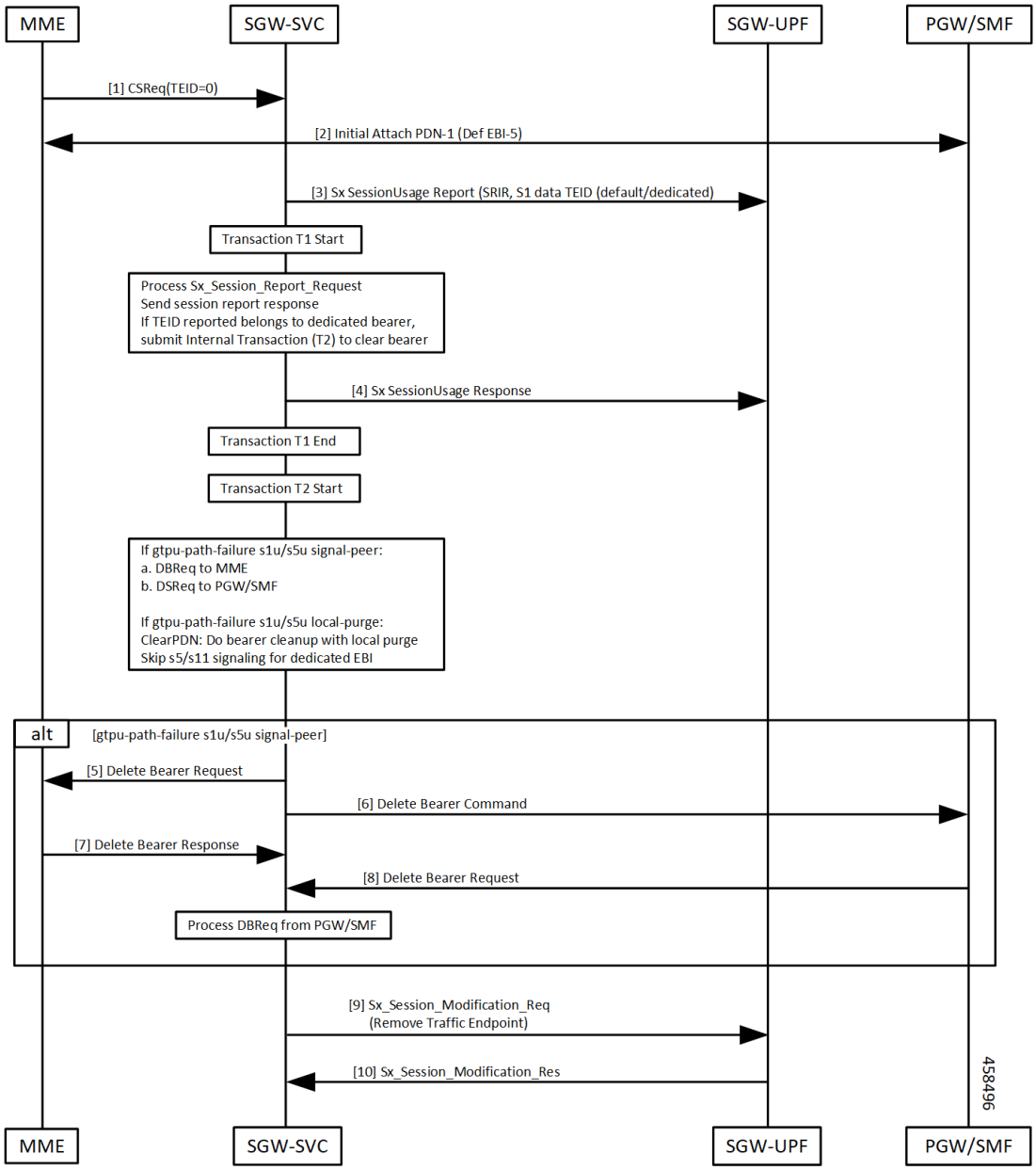


表 *105 :* 専用ベアラーのセッション置換コールフローの説明

ステップ	説明
1、2	初期接続が完了しました。

ステップ	説明
3、4	• UPF が、レポートタイプが SRIR および TEID の Sx_Session_Report_Request を送信します。 • cnSGW-C が Sx_Session_Report_Response で応答します。
5 ～ 10	cnSGW-C が Sx_Session_Report_Request を処理し、誤ったセッション置換により、ピアシグナリングまたはローカル消去のために GTPU パス障害 CLI が使用されます。 専用ベアラーの TEID を受信した場合は、ベアラーをクリーンアップするために、内部トランザクション（T2）を送信します。 CLI の場合、`gtpu-path-failure s1u/s5u signal-peer` • Sx_Session_Report_Request を UPF に送信し、FAR ACTION=DROP を設定します。 • Delete Bearer Req を MME に送信します。 • Delete Session Request を PGW に送信します。
11、12	• Sx_Session_Modification_Request（Remove Traffic Endpoint）を送信します。 • UPF が Sx_Session_Modification_Response で応答します。

機能設定

ここでは、GTPU エラー 表示機能を構成する方法について説明します。

この機能を設定するには、次の構成を使用します。

```
config
 profile sgw sgw_profile_name
  gtpu-error-ind
   s1u [ local-purge | page-ue ]
   s5u [ local-purge | signal-peer ]
   end
```

注：

- **s1u**：S1-U インターフェイス。
- **s5u**：S5-U インターフェイス。
- **local-purge**：ピアに通知せずに、影響を受けるベアラーまたは PDN をローカルに消去します。
- **page-ue**：S1 アイドル状態にリセットし、この UE のページングを開始します。

- **signal-peer**：ピアへのシグナリングで影響を受けるベアラーまたはPDNをクリアします。

設定例

次に、設定例を示します。

```
config
profile sgw sgw1
gtpu-error-ind s1u local-purge
gtpu-error-ind s5u signal-peer s1u local-purge
end
```

構成の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw gtpu-error-ind s1u local-purge
profile sgw sgw1
gtpu-error-ind s1u local-purge
```

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

次に、エラーレポートに起因する PDN クリーンアップの統計を示します。

```
sgw_pdn_disconnect_stats{app_name="smf",cluster="Local",data_center="DC",instance_id="0",

pdn_type="ipv4",rat_type="EUTRAN",reason="s1u_gtpu_error",service_name="sgw-service"}
 1
```

```
sgw_pdn_disconnect_stats{app_name="smf",cluster="Local",data_center="DC",instance_id="0",

pdn_type="ipv4",rat_type="EUTRAN",reason="s5u_gtpu_error",service_name="sgw-service"}
 1
```

```
sgw_pdn_disconnect_stats{app_name="smf",cluster="Local",data_center="DC",instance_id="0",

pdn_type="ipv4",rat_type="EUTRAN",reason="s1u_gtpu_session_replacement",service_name="sgw-service"}
 1
```

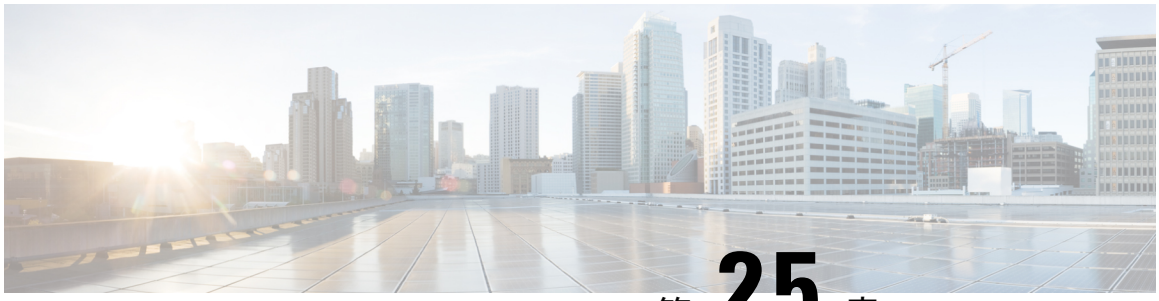
```
sgw_pdn_disconnect_stats{app_name="smf",cluster="Local",data_center="DC",instance_id="0",

pdn_type="ipv4",rat_type="EUTRAN",reason="userplane_requested_graceful_termination",service_name="sgw-service"}
 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",
sgw_procedure_type="s5u_gtpu_error_initiated_bearer_deletion",status="attempted",sub_fail_reason=""}
 1

sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",
sgw_procedure_type="s5u_gtpu_error_initiated_bearer_deletion",status="success",sub_fail_reason=""}
 1

sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",
sgw_procedure_type="s5u_gtpu_session_replacement_initiated_bearer_deletion",status="attempted",sub_fail_reason=""}
 1

sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_egress",reject_cause="",service_name="sgw-service",sgw_procedure_type=
"s5u_gtpu_session_replacement_initiated_bearer_deletion",status="success",sub_fail_reason=""}
 1

sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=
"s5u_gtpu_error_initiated_bearer_deletion",status="attempted",sub_fail_reason=""}
 1

sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=
 "s5u_gtpu_error_initiated_bearer_deletion",status="success",sub_fail_reason=""}
 1

sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=
"s5u_gtpu_session_replacement_initiated_bearer_deletion",status="attempted",sub_fail_reason=""}
 1

sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",gr_instance_id="1",
```

```
instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=

"s5u_gtpu_session_replacement_initiated_bearer_deletion",status="success",sub_fail_reason=""}
 1
```

S1u トンネルでの GTPU エラー表示が原因で開始された DDN に対して、次の SGW ddn_stats_type が追加されます。

```
sgw_ddn_stats{app_name="smf",cluster="cn",data_center="cn",ddn_stats_type="gtpu_err_ind_triggered",
 instance_id="0",service_name="sgw-service"} 2
```



第 25 章

GTPU パス障害

- 機能の概要と変更履歴 （299 ページ）
- 機能説明 （300 ページ）
- 機能の仕組み （300 ページ）
- 機能設定 （305 ページ）
- GTPU パス障害 OAM のサポート （306 ページ）

機能の概要と変更履歴

要約データ

表 *106* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	

マニュアルの変更履歴

表 *107* : マニュアルの変更履歴

改訂の詳細	リリース
ノードレポート（ノードIDとピア情報）で報告された値に基づくセッションまたはベアラーのクリーンアップのサポート	2021.02.3
最初の導入。	2021.02.1

機能説明

UPF は GTP-U パス障害を検出すると、ノードレポート要求（NodeID と GTPU ピア情報を含む）を cnSGW-C に送信します。cnSGW-C は、GTP-U ピアおよび UPF ノード ID に属する PDU セッションをクリアします。

この機能は次をサポートします。

- Node Report Success の送信
- node-report（ノード ID とピア情報）で報告された値に基づくセッションまたはベアラーのクリーニング
- 関連する統計の増加

機能の仕組み

ここでは、この機能の仕組みを説明します。

次の表では、GTPU パス障害の検出に関するさまざまなアクションについて説明します。

表 *108 :* ノードレポートの *GTPU* パス障害

インターフェイス	設定	TEID	Action
S1U/S5U	ローカル消去	デフォルト	SxSessionDletion を送信して UPF でクリーンアップ ローカルでの PDN のパージ
		Dedicated	SxSessionModification の送信（TrafficEndpointI の削除） ベアラーをローカルで消去
	シグナルピア	デフォルト	SxSessionModificationRequest（FAR アクション = DROP）を送信 DBReq を MME に、DSReq を PGW に送信 SxSessionDeletionRequest を送信
		Dedicated	DBReq を MME に、DBCmd を PGW に送信（非同期） Sx_Session_Modification_Request（Traffic Endpoint の削除）を送信

コール フロー

ここでは、GTPU パス障害の主要なコールフローについて説明します。

デフォルトのベアラーコールフローのパス障害

このセクションでは、デフォルのトベアラーコールフローのパス障害について説明します。

図 *53* : デフォルトのベアラーコールフローのパス障害

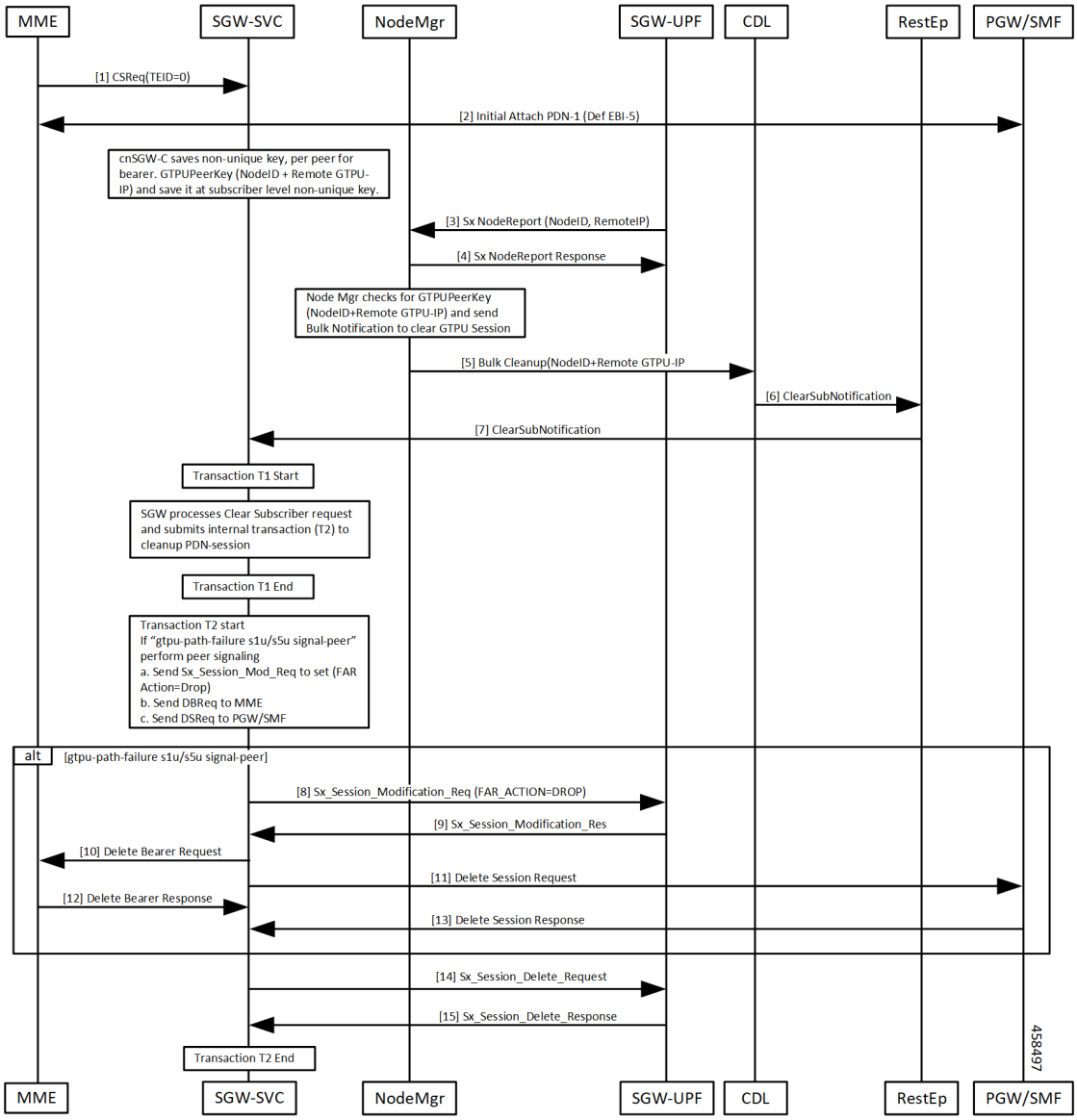


表 *109* : デフォルトのベアラコールフローのパス障害の説明

ステップ	説明
1、2	初期接続が完了しました。 cnSGW-C は、ベアラーごとに非一意の GTPUPeerKey（NodeID：Remote GTPU-peer-IP）を保存します。

ステップ	説明
3、4	• UPF でパス障害が検出されました。UPF が NodeReportRequest をノードマネージャに送信します。 • ノードマネージャは NodeReportResponse で応答します。 • CDL への一括クリーンアップ要求を開始します。
6、7	• CDL が ClearSubNotification を RestEp に送信します。 • RestEP はそれを cnSGW-C に転送します。
8–13	cnSGW-C が Clear Subscriber Request を処理します。 受信した GTPU ピア IP がデフォルトのベアラー用である場合は、PDN をクリーンアップするための内部トランザクション（T2）を実行します。 CLI の場合、`gtpu-path-failure s1u/s5u signal-peer` • Sx_Session_Report_Request を UPF に送信し、FAR ACTION=DROP を設定します。 • Delete Bearer Req を MME に送信します。 • セッション削除要求を PGW に送信します。
14、15	• Sx_Session_Delete_Request を送信します。 • UPF が Sx_Session_Delete_Response で応答します。

専用ベアラーのパス障害コールフロー

ここでは、専用ベアラーのパス障害コールフローについて説明します。

図 *54* : 専用ベアラーのパス障害コールフロー

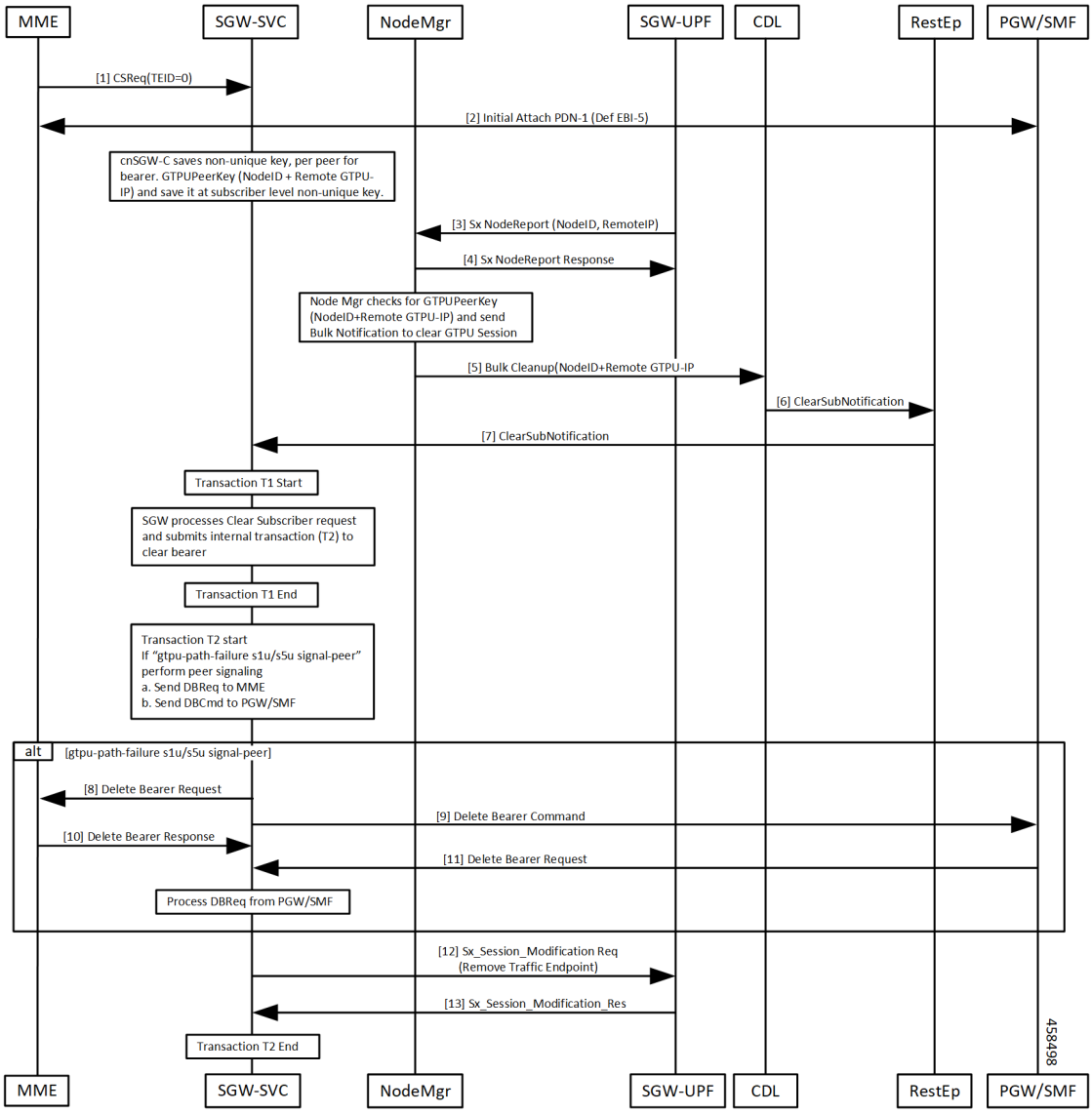


表 *110* : 専用ベアラーのパス障害コールフローの説明

ステップ	説明
1、2	初期接続が完了しました。cnSGW-C は、ベアラーごとに非一意の GTPUPeerKey（NodeID：Remote GTPU-peer-IP）を保存します。

ステップ	説明
3、4	• UPF でパス障害が検出されました。UPF が NodeReportRequest をノードマネージャに送信します。 • ノードマネージャは NodeReportResponse で応答します。 • CDL への一括クリーンアップ要求を開始します。
6、7	• CDL が ClearSubNotification を RestEp に送信します。 • RestEP はそれを cnSGW-C に転送します。
8 ～ 13	cnSGW-C がサブスクライバクリア要求を処理します。 GTPU ピア IP が専用ベアラー用に受信された場合は、内部トランザクション（T2）を送信して PDN をクリーンアップします。 CLI の場合、`gtpu-path-failure s1u/s5u signal-peer` • ベアラー削除要求を MME に送信します。 • ベアラー削除コマンドを PGW に送信します。
14、15	• Sx_Session_Modification_Request（トラフィックエンドポイントの削除）を送信します。 • UPF が Sx_Session_Delete_Response で応答します。

機能設定

ここでは、GTPU パス障害機能の設定方法について説明します。

この機能を設定するには、次の設定を使用します。

```
config
 profile sgw sgw_profile_name
  gtpu-path-failure
   s1u [ local-purge | signal-peer ]
   s5u [ local-purge | signal-peer ]
   end
```

注：

- **s1u**：S1-U インターフェイス。デフォルトは local-purge です。
- **s5u**：S5-U インターフェイス。デフォルトは local-purge です。

- **local-purge**：ピアに通知せずに、影響を受けるベアラーまたは PDN をローカルに消去します。
- **signal-peer**：ピアへのシグナリングで影響を受けるベアラーまたはPDNをクリアします。

設定例

次に、設定例を示します。

```
config
   profile sgw sgw1
   gtpu-path-failure s1u local-purge
   gtpu-path-failure s5u local-purge
   end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw sgw1
profile sgw sgw1
sgw-charging-threshold thre1
sgw-charging-profile ch1
locality LOC1
fqdn cisco.com.apn.epc.mnc456.mcc123
charging-mode gtpp
subscriber-policy sub1
session-idle-timer 86000
ddn failure-action-drop-timer 60
ddn no-user-connect-retry-timer 60
path-failure s11 signal-peer
path-failure s5e signal-peer
gtpu-error-ind s5u signal-peer
gtpu-path-failure s1u local-purge
gtpu-path-failure s5u local-purge
```

GTPU パス障害 OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

- ノードマネージャの統計

```
nodemgr_node_report_stats{app_name="smf",backlog_tmr="0",cluster="Local",data_center="DC",

instance_id="0",node_report_no_of_sess="0",node_report_peer_gtpu="209.165.201.20:209.165.201.30",

node_report_type="",service_name="nodemgr",session_tmr="0",status="attempted",
 up_ep_key="209.165.201.20:209.165.201.10"} 1
```

```
nodemgr_node_report_stats{app_name="smf",backlog_tmr="1617268831815934340",cluster="Local",
 data_center="DC",instance_id="0",node_report_no_of_sess="0",
node_report_peer_gtpu="209.165.201.20:209.165.201.30",node_report_type="origin",
 service_name="nodemgr",session_tmr="600",status="success",
up_ep_key="209.165.201.20:209.165.201.10"} 1
```

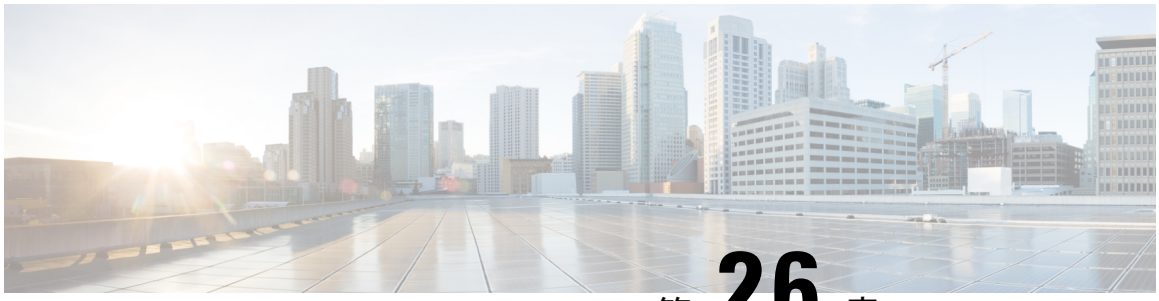
- SGW サービス統計

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

gr_instance_id="1",instance_id="0",interface="interface_sgw_egress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="s5u_gtpu_path_failure_initiated",
 status="attempted",sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

gr_instance_id="1",instance_id="0",interface="interface_sgw_egress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="s5u_gtpu_path_failure_initiated",
 status="success",sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

gr_instance_id="1",instance_id="0",interface="interface_sgw_ingress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="s5u_gtpu_path_failure_initiated",
 status="attempted",sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

gr_instance_id="1",instance_id="0",interface="interface_sgw_ingress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="s5u_gtpu_path_failure_initiated",
 status="success",sub_fail_reason=""} 1
```



第 26 章

GTPv2 および Sx メッセージの再送信およびタイムアウトの処理

- 機能の概要と変更履歴 （309 ページ）
- 機能説明 （310 ページ）
- 機能の仕組み （310 ページ）
- 再送信とタイムアウトの値の設定 （311 ページ）

機能の概要と変更履歴

要約データ

表 *111* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *112* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

cnSGW-C により、CLI を介して発信および着信メッセージに関連付けられたパラメータの再送信やタイムアウト処理が可能になります。再送信とタイムアウトの処理は、次のメッセージに適用されます。

- MME と PGW への GTPC インターフェイスを介したメッセージ
- ユーザープレーン（UP）に送信された Sx メッセージ

（注） 再送信およびタイムアウトのパラメータを処理するには、インターフェイス（S5e、S11、および Sxa）に再送信の設定（N3/T3）を追加する必要があります。

機能の仕組み

S-GW サービス、GTPC-EP、および SMF プロトコルは、再送信およびタイムアウトの処理に関与するプライマリノードです。

SGW サービスは以下に関与します。

- GTPC-EP および SMF プロトコルからのタイムアウトイベントの処理
- 再送信された着信メッセージの無視

GTPC-EP および SMF プロトコルは以下に関与します。

- 再送信およびタイムアウトの処理
- N3/T3 設定の読み取り
- 設定変更時の N3/T3 の更新

再送信とタイムアウトの処理は、発信メッセージと着信メッセージの両方に適用できます。

発信メッセージ

発信 GTP および PFCP メッセージの再送信とタイムアウトをサポートするには、ネットワークの応答時間や遅延時間に従って、インターフェイス固有の N3（最大再試行回数）タイマーと T3（再送信タイムアウト）タイマーの値を設定する必要があります。

MME/S11 ピアには、PGW/S5 または UPF/SXA とは異なる再送信タイムアウトを設定できます。

GTPC-EP/プロトコルポッドは、応答を受信するか、N3T3 が枯渇するまで、設定された N3T3 値に基づいて発信要求メッセージを再試行します。N3T3 が枯渇した場合、

GTPC-EP/プロトコルポッドは、発信要求メッセージに対する応答を受信していないことを示すために、ピアが応答しない原因に関する失敗応答をサービスポッドに送信します。

着信メッセージ

N4 および GTP エンドポイントごとに、着信および発信トラフィック用のキューのセットがあります。各キューでは、キューからメッセージをプルするディスパッチャスレッドが実行されます。メッセージは、追加処理のためにアプリケーションにディスパッチされます。

各ディスパッチャは、再送信キャッシュを参照して、着信要求がすでに使用されているか確認し、さらに、次のアクションを実行します。

- 再試行要求の場合、ディスパッチャは着信要求をドロップします。
- 再送信キャッシュが未処理の要求のしきい値に達すると、着信要求はドロップされます。

各ディスパッチャには、個別の再送信キャッシュがあります。このキャッシュは、送信された要求の応答でも更新され、応答を送信後に受信した再送信要求にも使用されます。

再送信とタイムアウトの値の設定

このセクションでは、発信メッセージと着信メッセージの再送信とタイムアウトの値を設定するための CLI コマンドについて説明します。

以下に、発信メッセージの CLI 設定を示します。

```
config
  instance instance-id instance_id
    endpoint endpoint_name
      interface interface_name
        retransmission timeout timeout_intervalmax-retry retry_value
        end
```

注：

- **instance instance-id** *instance_id*：インスタンス識別子を指定します。
- **endpoint** *endpoint_name*：エンドポイント名を指定します。
- **interface** *interface_name*：インターフェイス名を指定します。
- **retransmission timeout** *timeout_interval*：タイムアウト間隔の値を設定します。

以下に、着信メッセージの CLI 設定を示します。

```
config
    instance instance-id 1
    endpoint protocol
    interface n4
        dispatcher
```

```
        count 5
        outbound true
    threshold 5000
    end
```

注：

- **capacity** *capacity_value*：各ディスパッチャキューのキューサイズを指定します。デフォルト値は 5000 です。
- **count** *value*：インターフェイスまたはエンドポイントでサポートされるディスパッチャーキューの数を指定します。
- **expiry** *expiry_duration*：応答のあるキャッシュエントリがキャッシュに保持される期間を指定します。デフォルト値は 60 秒です。
- **nonresponsive** *nonresponsive_duration*：応答のないキャッシュエントリがキャッシュに保持される期間を指定します。
- **outbound** *true* | *false*：発信メッセージのディスパッチャキューのサポートを無効にします。デフォルト値は true です。false に設定すると、発信メッセージのキューサポートが有効になります。

  これは、デフォルトで発信メッセージのキューサポートが有効になっていることを意味します。以下のいずれかを指定する必要があります。

  - *true*：発信メッセージのディスパッチャキューのサポートを無効にします。**outbound** を true に設定します。
  - *false*：発信メッセージのディスパッチャキューのサポートを有効にします。**outbound** を false に設定します。

- **rate-limit** *rate_limit*：各キューのレート制限を指定します。
- **threshold** *threshold*：無応答キャッシュエントリの未処理の制限を指定します。しきい値に達すると、着信要求はドロップされます。整数である必要があります。デフォルト値は 30000 ミリ秒です。

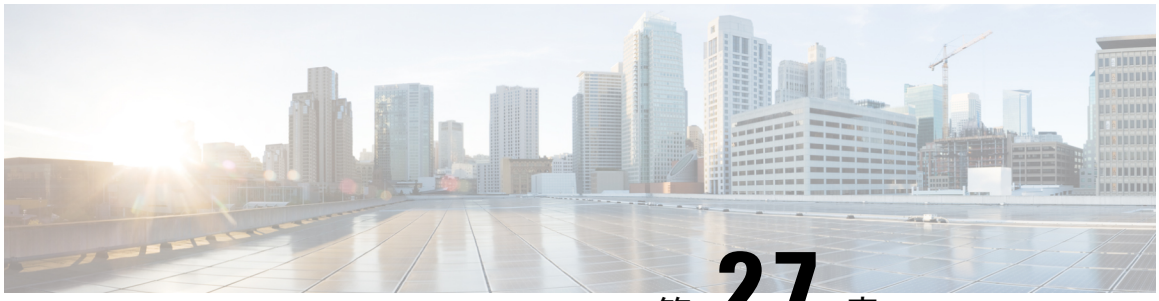
設定の確認

次に、発信メッセージと着信メッセージの再送信およびタイムアウト処理の構成を確認する構成例を示します。

```
show running-config instance instance-id 1 endpoint gtp
instance instance-id 1
endpoint gtp
replicas 1
interface s5e
retransmission timeout 2 max-retry 2
sla response 7000
dispatcher
count 1
capacity 1000
```

```
outbound true
threshold 10000
expiry 40000
nonresponsive 20000
exit
vip-ip 209.165.201.25
exit
interface s11
retransmission timeout 2 max-retry 2
sla response 7000
dispatcher
count 1
capacity 1000
outbound true
threshold 10000
expiry 40000
nonresponsive 20000
exit
vip-ip 209.165.201.2
exit
exit
```

```
show running-config instance instance-id 1 endpoint pfcp
instance instance-id 1
endpoint pfcp
replicas 1
interface sxa
retransmission timeout 2 max-retry 2
dispatcher
count 1
capacity 1000
outbound true
threshold 10000
expiry 40000
nonresponsive 20000
exit
heartbeat
interval 0
retransmission-timeout 3
max-retransmissions 5
exit
retransmission timeout 5 max-retry 1
exit
interface n4
heartbeat
interval 0
retransmission-timeout 3
max-retransmissions 5
exit
exit
exit
```



第 27 章

GTPv2 負荷/過負荷のサポート

- 機能の概要と変更履歴 （315 ページ）
- 機能説明 （316 ページ）
- GTPv2 負荷および過負荷機能の構成 （317 ページ）
- GTPv2 ロードおよびオーバーロード OAM のサポート （326 ページ）

機能の概要と変更履歴

要約データ

表 *113*：要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	無効：有効にするには設定が必要ですです
関連資料	該当なし

マニュアルの変更履歴

表 *114*：マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

次に、負荷制御機能と過負荷制御機能の詳細を示します。

負荷制御

負荷制御機能により、SGW や PGW などの GTPC エンティティは、MME、ePDG、TWAN などの GTPC ピアに負荷情報を送信できます。負荷情報は、有効負荷に応じて、SGW クラスタなどの同じ機能をサポートするすべてのノード間でセッションの負荷を分散するために使用されます。負荷情報には、GTPC エンティティリソースの動作ステータスが反映されます。

cnSGW-C の負荷制御動作は次のとおりです。

- CLI を使用して、cnSGW-C の負荷制御サポートをアクティブまたは非アクティブにします。
- 負荷制御機能がアクティブになると、cnSGW-C は次の理由でその負荷制御情報を MME に通知します。
  - 最適な GW の選択手順
  - ネットワーク内の cnSGW-C 全体のロードバランシングの強化
- 負荷制御情報の計算は、展開シナリオに基づいています。
- 該当する GTPC 要求または応答メッセージには、ピギーバックされた負荷制御情報が含まれています。
- cnSGW-C には、SGW 負荷制御情報に従って、LCI（負荷制御情報）IE の単一インスタンスのみが含まれます。cnSGW-C は、PGW から受信した LCI IE をその LCI 情報とともに MME に送信します。
- SGW に負荷制御情報を含める頻度は、展開シナリオに基づいています。SGW は、新規または更新された負荷制御情報のターゲットレシーバへの伝達が許容可能な遅延内に収まるようにします。この許容可能な遅延は、ネットワークで効果的なロードバランシング動作を実現するのに役立ちます。

過負荷制御

過負荷制御機能により、GTPC エンティティは、削減したトラフィックの送信を GTPC ピアに指示することで、自身の着信シグナリング負荷を正常に削減できます。GTPC エンティティがシグナリング容量を超えて動作すると、過負荷になるため、パフォーマンスが低下し、着信および発信トラフィックの処理に影響を与えます。GTPC ノードは、負荷情報を使用して、ノード間の GTPC シグナリングトラフィックの量を削減、スロットリング、または削減およびスロットリングします。

cnSGW-C の過負荷制御の動作は次のとおりです。

- CLI を使用して、cnSGW-C の過負荷制御サポートをアクティブまたは非アクティブにします。
- 過負荷制御機能がアクティブになると、cnSGW-C はその過負荷制御情報を MME または PGW に通知するため、自身への GTPC シグナリングトラフィックを制御するのに役立ちます。
- SGW は、該当するすべてのメッセージにおける過負荷制御情報の処理をサポートしています。
- 該当する GTPC 要求または応答メッセージには、ピギーバックされた過負荷制御情報が含まれています。
- cnSGW-C には、SGW 過負荷制御情報に従って、OCI（過負荷制御情報）IE の単一インスタンスのみが含まれます。cnSGW-C は、PGW から受信した OCI IE をその OCI 情報とともに MME に送信します。
- 過負荷制御情報の計算は、展開シナリオに基づいています。
- SGW が自己保護モードの場合、cnSGW-C は GTPC エンティティの輻輳を理由として拒否します。
- SGW は、MME や PGW 過負荷制御情報を保存しません。
- SGW は、MME および PGW に対するスロットリングを実行しません。

（注） 負荷制御および過負荷制御はオプションの機能です。

GTPv2 負荷および過負荷機能の構成

このセクションでは、GTPv2 負荷または過負荷状態を設定する方法について説明します。

この機能の設定には、次の手順が含まれます。

- 負荷プロファイルの設定：このセクションでは、負荷プロファイルと、cnSGW-C の負荷の計算に必要なパラメータを設定する方法について説明します。
- オーバーロード除外プロファイルの設定：このセクションでは、過負荷状態における除外設定の方法と、除外プロファイルの設定について説明します。
  - このプロファイルは、スロットリングの判断から除外するセッション関連メッセージを決定します。
  - 自己保護とピア過負荷制御の両方で、この設定を使用します。
- 過負荷条件プロファイルの構成（320 ページ）：このセクションでは、過負荷状態でのプロファイルの設定方法について説明します。

- プロファイルは、過負荷制御のさまざまな状態と、結果として生じるスロットリングの判断を決定します。
- サポートされる過負荷プロファイルは 1 つのみです。
- 負荷プロファイルは、過負荷プロファイル機能をサポートしています。

- 最大セッション数の設定 （321 ページ）：このセクションでは、LCI/OCI の計算においてセッション負荷率に寄与する最大セッション数の設定方法について説明します。
- 過負荷プロファイルと SGW プロファイルの関連付け （322 ページ）：過負荷プロファイルと SGW プロファイルの関連付けを設定できます。
- 負荷係数頻度の設定：このセクションでは、負荷係数の頻度を設定する方法を手順を追って説明します。

負荷プロファイルの構成

この機能を設定するには、次の設定を使用します。

```
config
   profile load profile_name
      load-calc-frequency load_calc_frequency_value
      load-fetch-frequency load_fetch_frequency_value
      advertise
      interval interval_value
      change-factor change_factor_value
      exit
   interface gtpc
      action advertise
end
```

注：

- **profile load** *profile_name*：負荷プロファイル名を指定します。
- **load-calc-frequency** *load_calc*_frequency_value：システムの負荷計算時間を秒単位で指定します。5 ～ 3600 の範囲の整数である必要があります。デフォルト値は 10 秒です。
- **load-fetch-frequency** *load_fetch*_frequency_value：プロトコルポッドがキャッシュポッドから負荷を取得する時間間隔を秒単位で指定します。5 ～ 3600 の範囲の整数である必要があります。デフォルト値は 10 秒です。
- **advertise interval** *interval_value*：ピアに LCI を送信する時間間隔を秒単位で指定します。0 ～ 3600 の範囲の整数で指定する必要があります。デフォルト値は 300 秒です。
- **advertise change-factor** *change_factor_value*：現在のロード値と最後に示されたロード値の差が変更係数を上回る場合は、対応するピアに LCI 値を指定します。1 ～ 20 の範囲の整数で指定する必要があります。デフォルト値は 5 です。
- **interface gtpc action advertise**：GTPC インターフェイスでの LCI 通知を有効にします。

設定例

次に、設定例を示します。

```
config
profile load pl
load-calc-frequency 30
load-fetch-frequency 60
advertise
interval 300
change-factor 1
exit
interface gtpc
action advertise
end
```

過負荷除外プロファイルの設定

デフォルトの動作の自己保護モードでは、cnSGWc はすべての着信 GTPC メッセージをスロットリングします。ただし、優先順位の高いメッセージをスロットリングすることはできないため、過負荷シナリオにおいても、cnSGWc はそれらの高優先順位のメッセージを除外する必要があります。過負荷除外プロファイルを設定すると、優先順位の高い着信メッセージをスロットリングから除外するインターフェイスを定義できます。

始める前に

負荷プロファイルを設定します。

手順

ステップ 1 **profile overload-exclude***overload_exclude_profile_name* コマンドを使用して、グローバル コンフィギュレーション モードで過負荷除外プロファイルインスタンスを設定します。

例：

```
[sgw] smf#config
[sgw] smf(config)#profile overload-exclude excludeProf
[sgw] smf(config-overload-exclude-excludeProf)#
```

ステップ 2 スロットリングの対象からメッセージを除外するために、**{ dnn-list** *dnn_list* **| arp-list** *arp_list* **| qci-list** *qci_list* **| message-priority { any | s5e | s11 | sxa } upto** *priority_val* **}**コマンドを使用して、DNN、ARP、QCI、メッセージ優先度のいずれかの過負荷除外プロファイルパラメータを定義します。

例：

```
[sgw] smf(config-overload-exclude-excludeProf)# dnn-list [ sos ]
[sgw] smf(config-overload-exclude-excludeProf)# arp-list [ 2 3 4 ]
[sgw] smf(config-overload-exclude-excludeProf)# qci-list [ 1 4 9 ]
[sgw] smf(config-overload-exclude-excludeProf)# message-priority s11 upto 5
```

次のタスク

過負荷除外プロファイルを設定後、次のタスクを実行する必要があります。

- 過負荷条件プロファイルを設定します。
- 最大セッション数を設定します。
- 過負荷除外プロファイルを cnSGWc プロファイルに関連付けます。
- 負荷係数頻度の計算を設定します。

過負荷条件プロファイルの構成

この機能を設定するには、次の構成を使用します。

```
config
   profile overload overload_profile_name
      overload-exclude-profile self-protection self_protection_profile_name
         node-level
         tolerance
            minimum min_percentage
            maximum max_percentage
         reduction-metric
            minimum min_percentage
            maximum max_percentage
         interface gtpc
            overloaded-action advertise
               advertise
                  interval interval_value
                  change-factor change_factor_value
                  validity-period validity_period_value
                  end
```

注：

- **profile overload** *overload_profile_name*：過負荷プロファイル名を指定します。
- **overload-exclude-profile self-protection** *self_protection_profile_name*：（これは任意の設定です）自己保護状態でのスロットリングの判断からメッセージを除外します。
- **tolerance minimum** *min_percentage* **maximum** *max_percentage*：システムの過負荷制限を指定します。次のシナリオを参照してください。
  - システムの負荷が *min_percentage* 未満の場合、システムは正常な状態です。
  - システムの負荷が *min_percentage* と *max_percentage* の間にある場合、システムは過負荷状態です。このシナリオでは、ノードの過負荷制御アクションがトリガーされます。
  - システムの負荷が *max_percentage* よりも大きい場合、システムは自己保護状態になります。このシナリオでは、自己保護アクションがトリガーされます。

- *max_percentage* は 1 ～ 100 の範囲の整数で指定する必要があります。デフォルト値は 95 です。
- *min_percentage* は 1 ～ 100 の範囲の整数で指定する必要があります。デフォルト値は 80 です。

- **reduction-metric minimum** *min_percentage* **maximum** *max_percentage*：削減メトリックの制限を指定します。次のシナリオを参照してください。
  - *min_percentage* と *max_percentage* のパーセンテージ値はどちらも **tolerance** 設定と連動します。
  - パーセンテージ値 *max_percentage* は、1 ～ 100 の範囲の整数で指定する必要があります。デフォルト値は 100 です。
  - パーセンテージ値 *min_percentage* は、1 ～ 100 の範囲の整数で指定する必要があります。デフォルト値は 10 です。

  例：次の条件では、負荷が 80% の場合は 10% の OCI をピアノードに送信し、負荷が 95% の場合は 30% の OCI をピアノードに送信します。
  - **tolerance** *min_percentage* は 80、*max_percentage* は 95 です。
  - **reduction-metric** *min_percentage* は 10、*max_percentage* は 30 です。

- **interface gtpc overloaded-action advertise**：ノードが過負荷になった場合の GTPC インターフェイスでのアクションを設定します。GTPC には、S5/S8/S11/S2b インターフェイスが含まれます。一部のアクションは、特定のインターフェイスにのみ適用されます。
- **advertise interval** *interval_value*：ピアに LCI を送信する周期を秒単位で指定します。0 ～ 3600 の範囲の整数で指定する必要があります。デフォルト値は 300 秒です。
- **advertise change-factor** *change_factor_value*：変更係数の値を指定します。現在の負荷値と最後に示された負荷値との差が変更係数の値を上回る場合は、GTPC は対応するピアに LCI を送信します。1 ～ 20 の範囲の整数で指定する必要があります。デフォルト値は 5 です。
- **advertise validity-period** *validity_period_value*：アドバタイズされた OCI 値の有効期間を秒単位で指定します。1 ～ 3600 の範囲の整数で指定する必要があります。デフォルト値は 600 秒です。

最大セッション数の設定

この機能を設定するには、次の設定を使用します。

```
config
   profile converged-core profile_name
      max-session-count max_session_count_value
      end
```

注：

- **profile converged-core** *profile_name*：コンバージドコア プロファイルの名前を指定します。
- **max-session-count** *max_session_count_value*：サポートされるセッションの最大数を指定します。1 ～ 12000000 の範囲の整数で指定する必要があります。

設定例

次に、設定例を示します。

```
config
profile converged-core convergedCoreProfile
max-session-count 12000000
exit
```

過負荷プロファイルと SGW プロファイルの関連付け

過負荷プロファイルと SGW プロファイルの関連付けを構成できます。

この機能を設定するには、次の構成を使用します。

```
config
   profile overload overload_profile_name
      overload-exclude-profile self-protection self_protection_profile_name
   node-level
      tolerance
         minimum min_percentage
         maximum max_percentage
      reduction-metric
         minimum min_percentage
         maximum max_percentage
         advertise
         interval interval_value
         change-factor
         exit
      interface gtpc
         overloaded-action [ advertise ]
         exit
      exit
   exit
   profile load load_name
   load-calc-frequency load_calc_frequency_value
   load-fetch-frequency load_fetch_frequency_value
   advertise
   interval interval_value
   change-factor change_factor_value
```

```
    exit
    interface gtpc
    action advertise
    exit
exit
profile sgw sgw_name
load-profile profile_name
overload-profile overload_profile_name
end
```

注：

- **overload** *overload_name*：過負荷保護プロファイル名を指定します。文字列で指定する必要があります。
- **overload-exclude-profile**：過負荷シナリオのプロファイルを除外します。
- **self-protection** *overload_value*：自己保護のために除外するプロファイルを指定します。文字列で指定する必要があります。
- **tolerance minimum** *min_percentage*：最小許容レベルを指定します（この値を下回ると、システムは正常状態になります）。1～100の範囲の整数で指定する必要があります。デフォルト値は 80 です。
- **tolerance maximum** *max_percentage*：最大許容レベルを指定します（この値を上回ると、システムは自己保護状態になります）。1～100の範囲の整数で指定する必要があります。デフォルト値は 95 です。
- **reduction-metric minimum** *min_percentage*：構成の最小許容レベルとともに削減率を指定します。1 ～ 100 の範囲の整数で指定する必要があります。デフォルト値は 10 です。
- **reduction-metric maximum** *max_percentage*：構成の最大許容レベルとともに削減率を指定します。1 ～ 100 の範囲の整数で指定する必要があります。デフォルト値は 100 です。
- **interval** *interval_value*：アドバタイズ間隔を秒単位で指定します。0 ～ 3600 の範囲の整数で指定する必要があります。デフォルト値は 300 秒です。
- **validity** *validity_value*：アドバタイズされた OCI 値の有効期間を秒単位で指定します。1 ～ 3600 の範囲の整数で指定する必要があります。デフォルト値は 600 秒です。
- **change-factor** *change_factor_value*：現在の OCI と最後に指定された OCI との最小変更値を指定します。この値を超えると、アドバタイズが行われます。1～20の範囲の整数で指定する必要があります。デフォルト値は 5 です。
- **profile load** *load_name*：負荷プロファイルの名前を指定します。文字列で指定する必要があります。
- **load-calc-frequency** *load_calc_frequency_value*：システム負荷計算間隔を秒単位で指定します。5 ～ 3600 の範囲の整数で指定する必要があります。デフォルト値は 10 秒です。
- **load-fetch-frequency** *load_fetch_frequency_value*：サービスポッドがキャッシュポッドから負荷をフェッチする時間間隔を秒単位で指定します。5～3600の範囲の整数で指定する必要があります。デフォルト値は 10 秒です。

- **load-profile** *profile_name*：負荷プロファイルの名前を指定します。
- **overload-profile** *overload_profile_name*：過負荷プロファイルの名前を指定します。
- ：除外過負荷プロファイル名を指定します。
  - ：スロットリングの決定から除外する必要がある ARP リストを指定します。1 ～ 15 の範囲の整数で指定する必要があります。最大 8 個のエントリが許可されます。
  - ：スロットリングの決定から除外する必要がある DNN のリストを指定します。最大 3 個のエントリが許可されます。
  - **message-priority**：メッセージの優先順位のどこまでをスロットリングの決定から除外するかを指定します。
  - **procedure-list**：スロットリングの決定から除外する手順。このパラメータは、自己保護にのみ適用されます。
  - ：スロットリングの決定から除外する QoS クラス識別子を指定します。1 ～ 254 の範囲の整数で指定する必要があります。最大 8 個のエントリが許可されます。たとえば、範囲の値は 1-9、65、66、69、70、80、82、83、128-254 です。

設定例

次に、設定例を示します。

```
config
profile overload op
overload-exclude-profile self-protection <overload-exclude-profile-name>
node-level
tolerance minimum 5
tolerance maximum 50
reduction-metric minimum 50
reduction-metric maximum 100
advertise
interval 0
change-factor 1
exit
interface gtpc
overloaded-action [ advertise ]
exit
exit
exit
profile load lp
load-calc-frequency 120
load-fetch-frequency 15
advertise
interval 0
change-factor 1
exit
interface gtpc
action advertise
exit
exit
profile sgw <sgw_name>
load-profile <profile_name>
overload-profile <overload_profile_name>
end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile
profile sgw sgw1
load lp1
overload op1
end
```

負荷係数パラメータの構成

cnSGWc は、特定の時間間隔でシステムの負荷係数を計算し、ピアノードにアドバタイズします。ただし、この構成では、計算から除外する非参加ポッドとともに、負荷係数の計算頻度を指定できます。負荷係数の計算が構成されていない場合、デフォルトでは、cnSGWc はすべての app-infra ポッドを除外して、30 秒間隔で負荷係数を計算します。

手順

ステップ 1 コマンド **load factor**を使用して、負荷係数の計算頻度を設定します。

例：

```
[sgw] smf(config)# load factor
```

ステップ 2 コマンド **[no] load factor { calc-frequency** *calc_ frequency_time* **| exclude-pods** *exclude_pod_name***}** を使用して、負荷係数の計算頻度や除外する app-infra ポッドを定義します。

例：

```
[sgw] smf(config)# load factor calc-frequency 15
[smf] smf(config)# load factor exclude-pods bgpspeaker-pod
```

- CLI **calc-frequency** のデフォルト値は 30 です。
- CLI **exclude-pods** により、次の app-infra ポッドを除外します。
  - **bfdmgr**
  - **bgpspeaker-pod**
  - **cache-pod**
  - **edr-monitor**
  - **georeplication-pod**
- CLI **calc-frequency** と **exclude-pods** はどちらも下位互換性があります。

ステップ 3 コマンド **exit** を使用してグローバル コンフィギュレーション モードを閉じます。

例：

```
[sgw] smf(config)#exit
```

設定の確認

show running-config load コマンドは、負荷係数の構成パラメータを表示します。

```
[sgw] smf# show running-config load
  load factor calc-frequency 30
  load factor exclude-pods bgpspeaker-pod
exit
```

GTPv2 ロードおよびオーバーロード OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

標準

```
node_overload_status{app_name="smf", cluster="cn", data_center="cn",
instance_id="0", service_name="oam-pod"} 0
```

Overload

```
node_overload_status{app_name="smf", cluster="cn", data_center="cn",
instance_id="0", service_name="oam-pod"} 1
```

自己保護状態

```
node_overload_status{app_name="smf", cluster="cn", data_center="cn",
instance_id="0", service_name="oam-pod"} 2
```

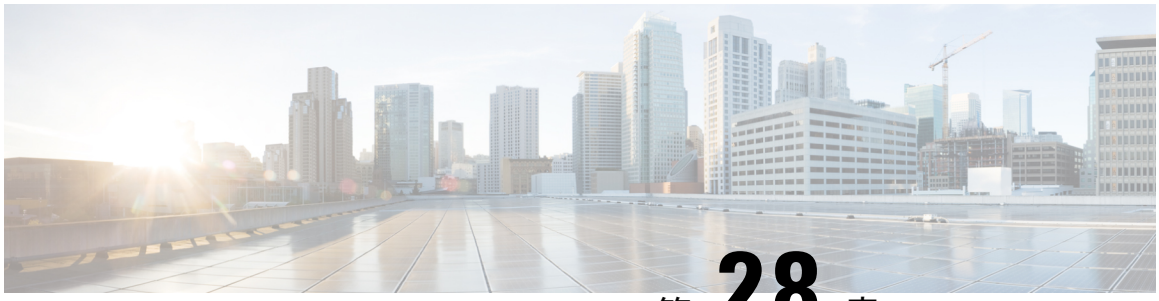
SGW サービス統計情報

```
sgw_service_stats{app_name="smf", cluster="cn", data_center="cn",
fail_reason="gtp_entity_in_congestion", instance_id="0",
interface="interface_sgw_ingress", reject_cause="entity_in_congestion",
service_name="sgw-service", sgw_procedure_type="initial_attach",
status="rejected", sub_fail_reason=""}
```

LCI/OCI メトリック値

```
node_lci_metric{app_name="SGW", cluster="cn", component="oam-pod",
data_center="DC", namespace="cn", instance_id="0", service_name="oam-pod"}
```

```
node_oci_metric{app_name="SGW", cluster="cn", component="oam-pod",
data_center="DC", namespace="cn", instance_id="0", service_name="oam-pod"}
```



第 28 章

GTPv2 メッセージ検証

- 機能の概要と変更履歴 （329 ページ）
- 機能説明 （330 ページ）
- 機能の仕組み （330 ページ）

機能の概要と変更履歴

要約データ

表 *115* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *116* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

cnSGW-C は、IE（値、必須 IE、およびサービス依存 IE）の基本的な GTPv2 メッセージ検証をサポートし、SGW サービス/GTPC-EP ポッドから応答を送信します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

SGW 入力（S11）の基本検証と事前検証コールフロー

ここでは、SGW 入力（S11）の基本検証と事前検証コールフローについて説明します。

図 *55 :* 基本検証と事前検証：*S11* コールフロー

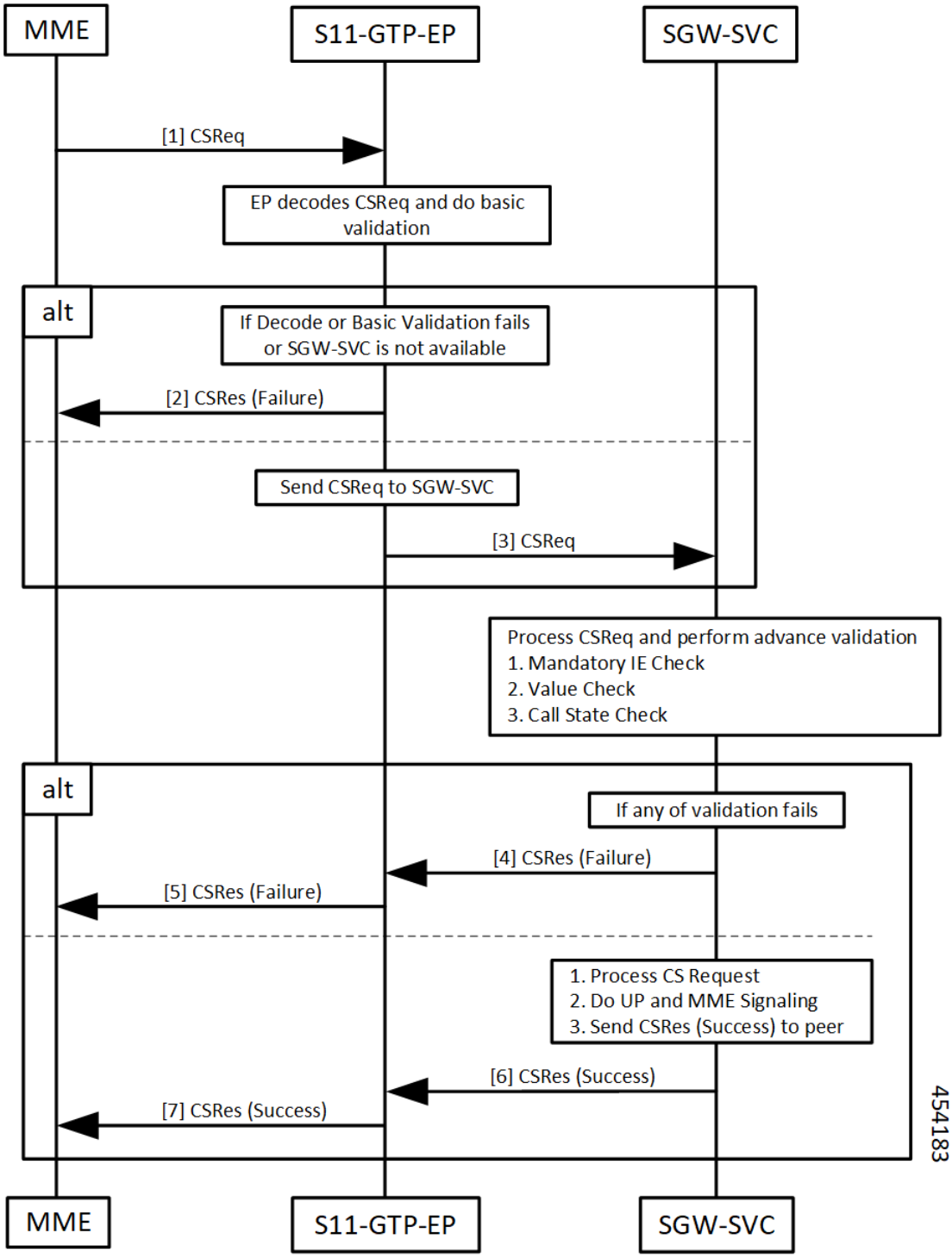


表 *117 :* 基本検証と事前検証：*S11* コールフローの説明

ステップ	説明
1	MME が Create Session Request を S11-GTP-EP ポッドに送信します。 S11-GTP-EP ポッドが Create Session Request を復号化し、基本検証を実行します。
2	復号化または基本検証が失敗した場合、あるいは SGW-SVC が使用できない場合、S11-GTP-EP ポッドが Create Session Response 失敗メッセージを MME に送信します。
3	Create Session Request の基本検証が成功した場合、S11-GTP-EP ポッドが Create Session Request を SGW-SVC ポッドに転送します。 SGW-SVC が Create Session Request を処理し、次の手順を実行します。 • 必須の IE チェック • 値チェック • コール状態のチェック
4、5	ステップ 3 の検証が失敗した場合： • SGW-SVC が Create Session Response 失敗メッセージを S11-GTP-EP ポッドに送信します。 • S11-GTP-EP ポッドが Create Session Response 失敗メッセージを MME に転送します。
6、7	ステップ 3 の検証が成功した場合、SGW-SVC が次の手順を実行します。 • Create Session Request の処理 • UP および MME シグナリングの実行 • Create Session Response 成功メッセージが S11-GTP-EP ポッドに送信されます。 S11-GTP-EP ポッドが Create Session Response 成功メッセージを MME に転送します。

SGW 出力（S5）の基本検証と事前検証コールフロー

ここでは、SGW 出力（S5）の基本検証および事前検証コールフローについて説明します。

図 *56* : 基本検証と事前検証：*S5* コールフロー

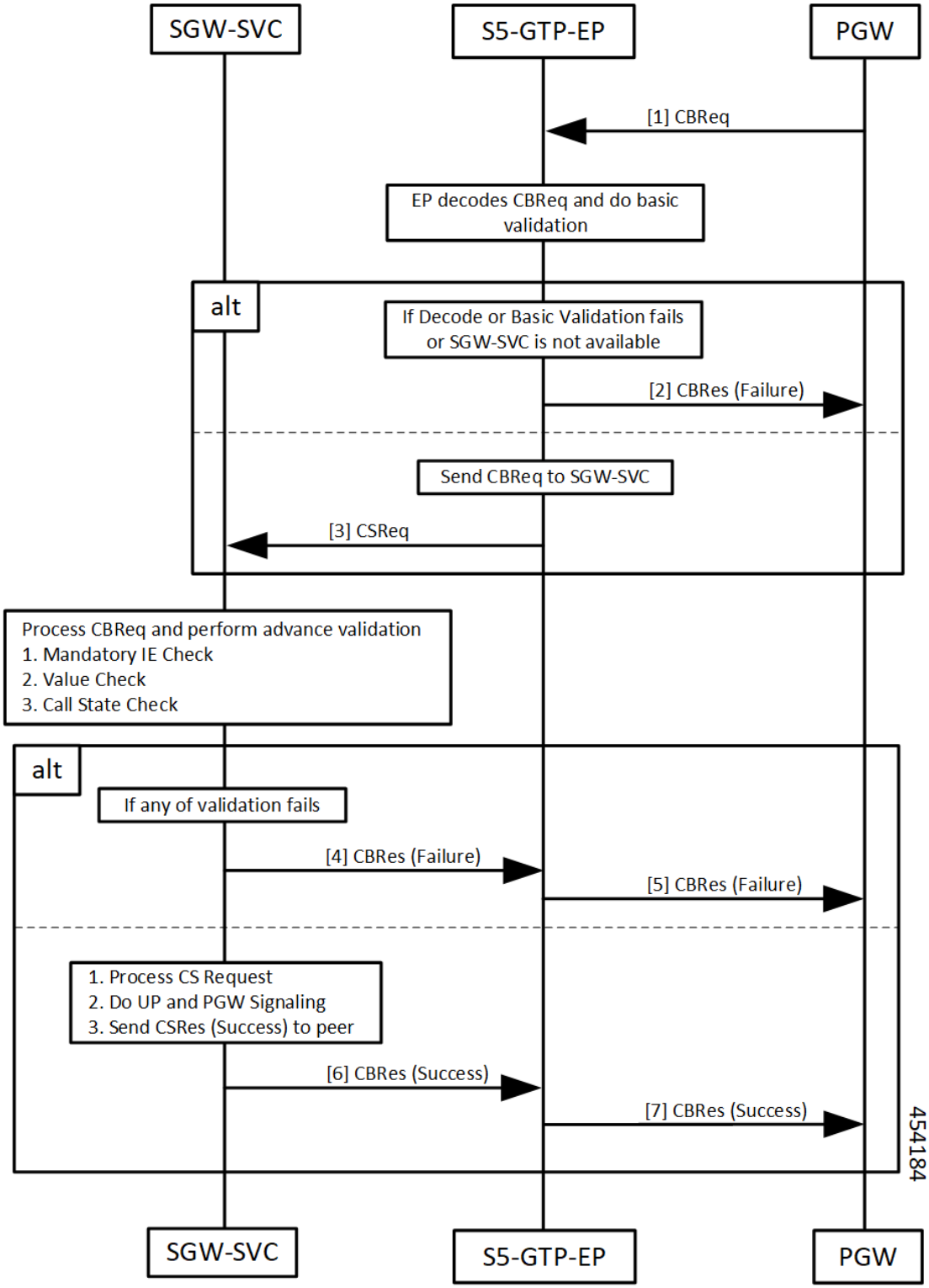
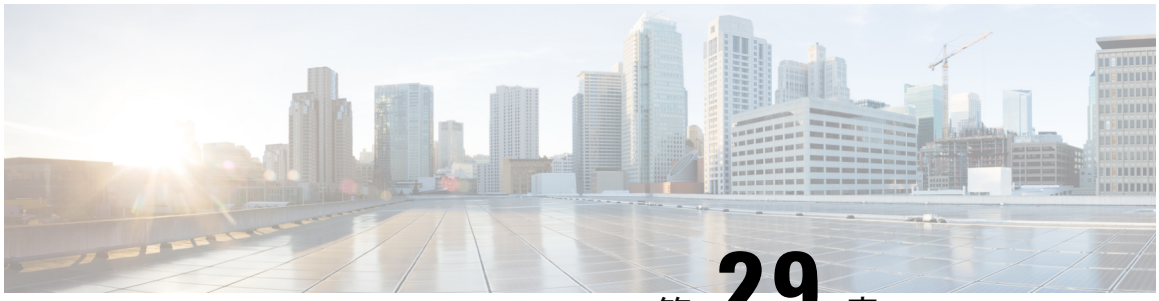


表 *118 :* 基本検証と事前検証：*S5* コールフローの説明

ステップ	説明
1	PGW が Create Bearer Request を S5-GTPC-EP ポッドに送信します。 S5-GTPC-EP ポッドが Create Bearer Request を復号化し、基本検証を実行します。
2	復号化または基本検証が失敗した場合、あるいは SGW-SVC が使用できない場合、S5-GTP-EP ポッドが Create Bearer Response 失敗メッセージを PGW に送信します。
3	Create Bearer Request の基本検証が成功した場合、S5-GTP-EP ポッドが Create Bearer Request を SGW-SVC ポッドに転送します。 SGW-SVC が Create Bearer Request を処理し、次の手順を実行します。 • 必須の IE チェック • 値チェック • コール状態のチェック
4、5	ステップ 3 の検証が失敗した場合： • SGW-SVC が Create Bearer Response 失敗メッセージを S5-GTP-EP ポッドに送信します。 • S5-GTP-EP ポッドが Create Bearer Response 失敗メッセージを PGW に転送します。
6、7	ステップ 3 の検証が成功した場合、SGW-SVC が次の手順を実行します。 • Create Bearer Request の処理 • UP および PGW シグナリングの実行 • Create Bearer Response 成功メッセージを S5-GTP-EP ポッドに送信します。 S5-GTP-EP ポッドが Create Bearer Response 成功メッセージを PGW に転送します。



第 29 章

IDFT サポート

- 機能の概要と変更履歴 （335 ページ）
- 機能説明 （336 ページ）
- 機能の仕組み （336 ページ）
- OAM サポート （346 ページ）

機能の概要と変更履歴

要約データ

表 *119* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *120* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

cnSGW-C は、SGW 再配置の有無にかかわらず、専用ベアラを使用した Pure-S コールの間接転送トンネル（IDFT）の作成と削除をサポートします。

標準準拠

この機能は、次の標準仕様に準拠しています。

- 3GPP TS 23.401「General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access」
- 3GPP TS 23.402「Architecture enhancements for non-3GPP accesses」
- 3GPP TS 29.274「Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C)」
- 3GPP TS 23.214「Architecture enhancements for control and user plane separation of EPC nodes」
- 3GPP TS 29.244「Interface between the Control Plane and the User Plane nodes」
- 3GPP TS 24.008「Mobile radio interface Layer 3 specification; Core network protocols; Stage 3」

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

SGW 再配置を伴わない IDFT のサポートコールフロー

ここでは、SGW 再配置を伴わない IDFT のサポートコールフローについて説明します。

図 *57 : SGW* 再配置を伴わない *IDFT* のサポートコールフロー

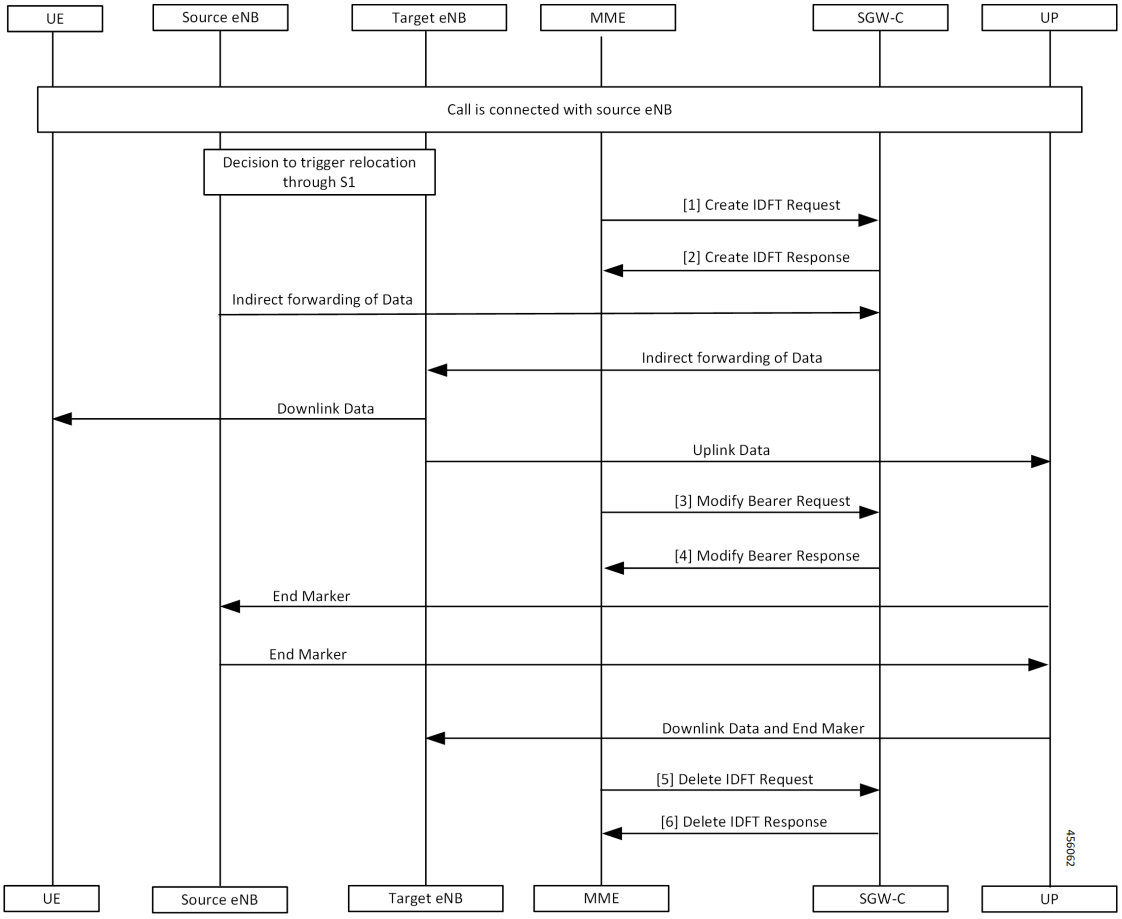


表 *121 : SGW* 再配置を伴わない *IDFT* のサポートコールフローの説明

ステップ	説明
1	コールが送信元 eNodeB に接続され、S1 経由で再配置をトリガーすることが決定されます。 MME が SGW-C に IDFT 作成要求を送信します。
2	MME が SGW-C から IDFT 作成応答を受信します。
3	データの間接転送が、送信元 eNodeB から SGW-C に向けて開始されます。 データの間接転送が、 SGW-C から eNodeB に向けて開始されます。 ターゲット eNodeB が UE にダウンリンクデータを送信します。 ターゲット eNodeB が UP にアップリンクデータを送信します。 MME が SGW-C にベアラー変更要求を送信します。

ステップ	説明
4	SGW-C が MME にベアラー変更応答を送信します。 UP が終了マーカーを送信元 eNodeB に送信し、送信元 eNodeB が終了マーカーを UP に転送します。 UP がダウンリンクデータと終了マーカーをターゲット eNodeB に送信します。
5	MME が SGW-C に IDFT 削除要求を送信します。
6	MME が SGW-C から IDFT 削除応答を受信します。

SGW 再配置を伴う IDFT のサポートコールフロー

ここでは、SGW 再配置を伴う IDFT のサポートコールフローについて説明します。

図 *58 : SGW* 再配置を伴う *IDFT* のサポートコールフロー

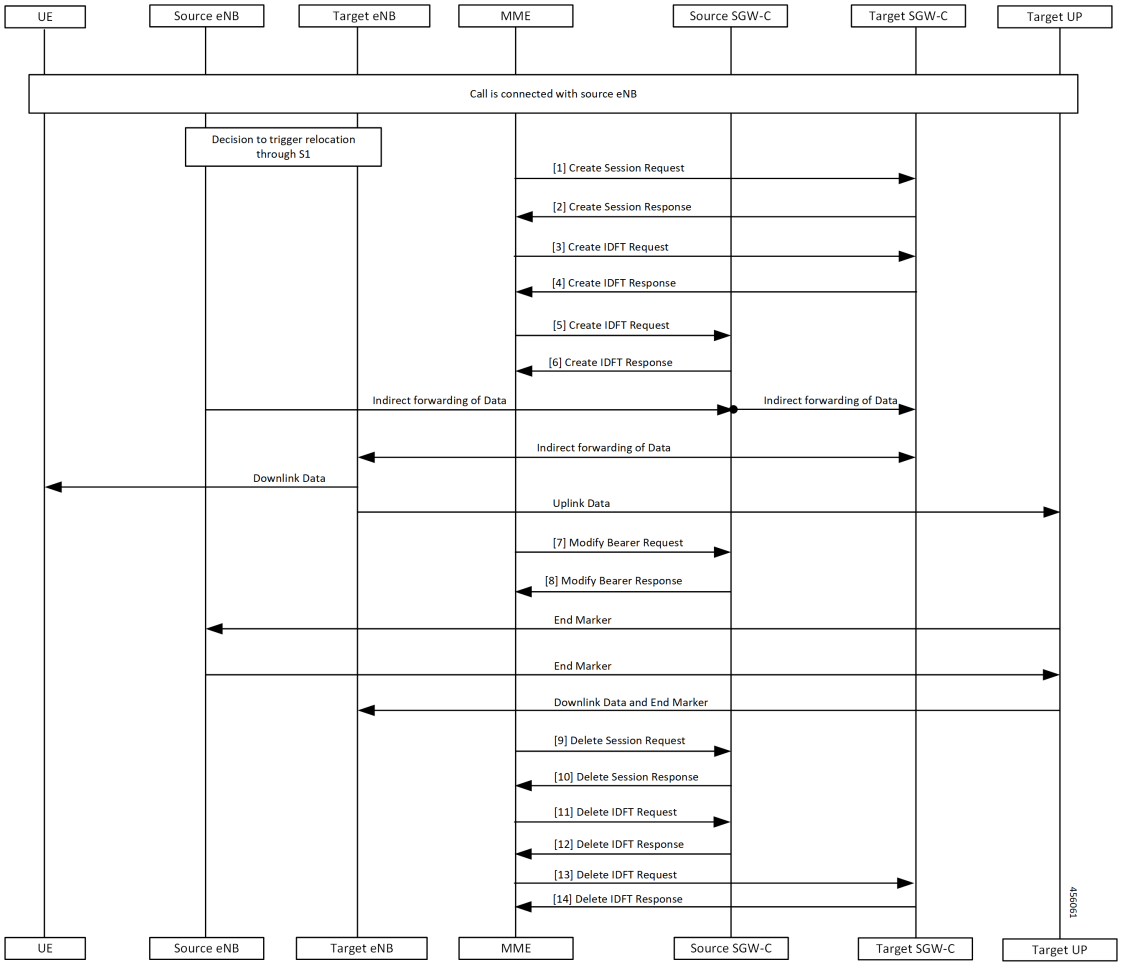


表 *122 : SGW* 再配置を伴う *IDFT* のサポートコールフローの説明

ステップ	説明
1	コールが送信元 eNodeB に接続され、S1 経由で再配置をトリガーすることが決定されます。 MME が ターゲット SGW-C にセッション作成要求を送信します。
2	MME がターゲット SGW-C からセッション作成応答を受信します。
3	MME がターゲット SGW-C に IDFT 作成要求を送信します。
4	MME がターゲット SGW-C から IDFT 作成応答を受信します。
5	MME が送信元 SGW-C に IDFT 作成要求を送信します。
6	MME が送信元 SGW-C から IDFT 作成応答を受信します。 データの間接転送が、送信元 eNodeB から送信元 SGW-C に向けて開始されます。 データの間接転送が、送信元 SGW-C からターゲット SGW-C に向けて開始されます。 データの間接転送が、ターゲット SGW-C からターゲット eNodeB に向けて開始されます。 ターゲット eNodeB が UE にダウンリンクデータを送信します。 ターゲット eNodeB がターゲット UP にアップリンクデータを送信します。
7	MME がターゲット SGW-C にベアラー変更要求を送信します。
8	送信元 SGW-C が MME にベアラー変更応答を送信します。 ターゲット UP が終了マーカーを送信元 eNodeB に送信し、送信元 eNodeB が終了マーカーをターゲット UP に転送します。 ターゲット UP がダウンリンクデータと終了マーカーをターゲット eNodeB に送信します。
9	MME が送信元 SGW-C にセッション削除要求を送信します。
10	MME が送信元 SGW-C からセッション削除応答を受信します。
11	MME が送信元 SGW-C に IDFT 削除要求を送信します。
12	MME がターゲット SGW-C から IDFT 削除応答を受信します。
13	MME がターゲット SGW-C に IDFT 削除要求を送信します。
14	MME がターゲット SGW-C から IDFT 削除応答を受信します。

Pure-S コールの 5G から 4G へのハンドオーバーのコールフロー

このセクションでは、Pure-S コールの 5G から 4G へのハンドオーバーのコールフローについて説明します。

図 *59 : Pure-S* コールの *5G* から *4G* へのハンドオーバーのコールフロー

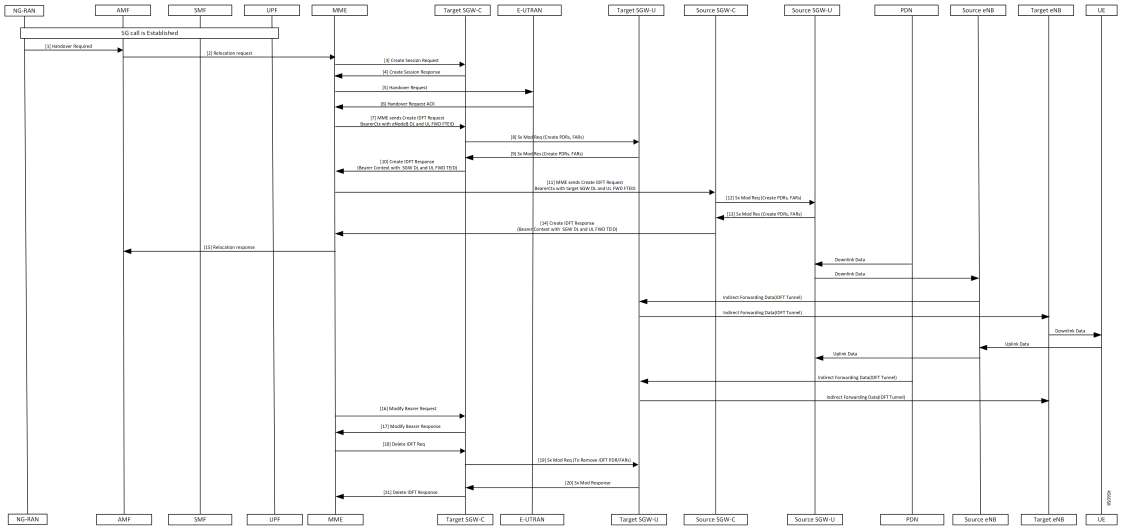


表 *123 : Pure-S* コールの *5G* から *4G* へのハンドオーバーのコールフローの説明

ステップ	説明
1	5G コールが確立されます。 NG-RAN が AMFにハンドオーバーが必要であることを示すメッセージを送信します。
2	AMF が再配置要求を MME に送信します。
3	MME がターゲット SGW-C にセッション作成要求を送信します。
4	MME がターゲット SGW-C からセッション作成応答を受信します。
5	MME が E-UTRAN にハンドオーバー要求を送信します。
6	MME が E-UTRAN からハンドオーバー要求の ACK を受信します。
7	MME が、eNodeB のダウンリンクおよびアップリンクの転送用 FTEID を含むベアラーコンテキストとともに、IDFT 作成要求をターゲット SGW-C に送信します。
8	ターゲット SGW-C が、PDR および FAR 作成を含む Sx 変更要求をターゲット SGW-U に送信します。

ステップ	説明
9	ターゲット SGW-U が、PDR および FAR 作成を含む Sx 変更応答をターゲット SGW-C に送信します。
10	ターゲット SGW-C が、SGW ダウンリンクおよびアップリンク転送用 TEID を含むベアラーコンテキストとともに IDFT 作成応答を MME に送信します。
11	MME が、SGW ダウンリンクおよびアップリンク転送用 FTEID を含むベアラーコンテキストとともに IDFT 作成要求を送信元 SGW-C に送信します。
12	送信元 SGW-C が、PDR および FAR 作成を含む Sx 変更要求を送信元 SGW-U に送信します。
13	送信元 SGW-C が、PDR および FAR 作成を含む Sx 変更応答を送信元 SGW-U から受信します。
14	MME が、SGW ダウンリンクおよびアップリンク転送用 TEID を含むベアラーコンテキストとともに、IDFT 作成応答を送信元 SGW-C から受信します。
15	MME が再配置応答を AMF に送信します。 PDN がダウンリンクデータを送信元 SGW-U に送信し、送信元 SGW-U はダウンリンクデータを送信元 eNodeB に送信します。 間接転送データ（IDFT トンネル）は送信元 eNodeB からターゲット SGW-U へ、およびターゲット SGW-U からターゲット eNodeB への方向で始まります。 ターゲット eNodeB が UE にダウンリンクデータを送信します。 UE がアップリンクデータを送信元 eNodeB に送信し、送信元 eNodeB はアップリンクデータを送信元 SGW-U に送信します。 PDN は間接転送データをターゲット SGW-U に送信し、ターゲット SGW-U は間接転送データをターゲット eNodeB に送信します。
16	MME がターゲット SGW-C にベアラー変更要求を送信します。
17	MME がターゲット SGW-C からベアラー変更応答を受信します。
18	MME がターゲット SGW-C に IDFT 削除要求を送信します。
19	ターゲット SGW-C が、PDR および FAR（削除対象）を含む Sx 変更要求をターゲット SGW-U に送信します。
20	ターゲット SGW-C が Sx 変更応答をターゲット SGW-U から受信します。
21	MME がターゲット SGW-C から IDFT 削除応答を受信します。

Pure-S コールの 4G から 5G へのハンドオーバーのコールフロー

このセクションでは、Pure-S コールの 4G から 5G へのハンドオーバーのコールフローについて説明します。

図 *60 : Pure-S* コールの *4G* から *5G* へのハンドオーバーのコールフロー

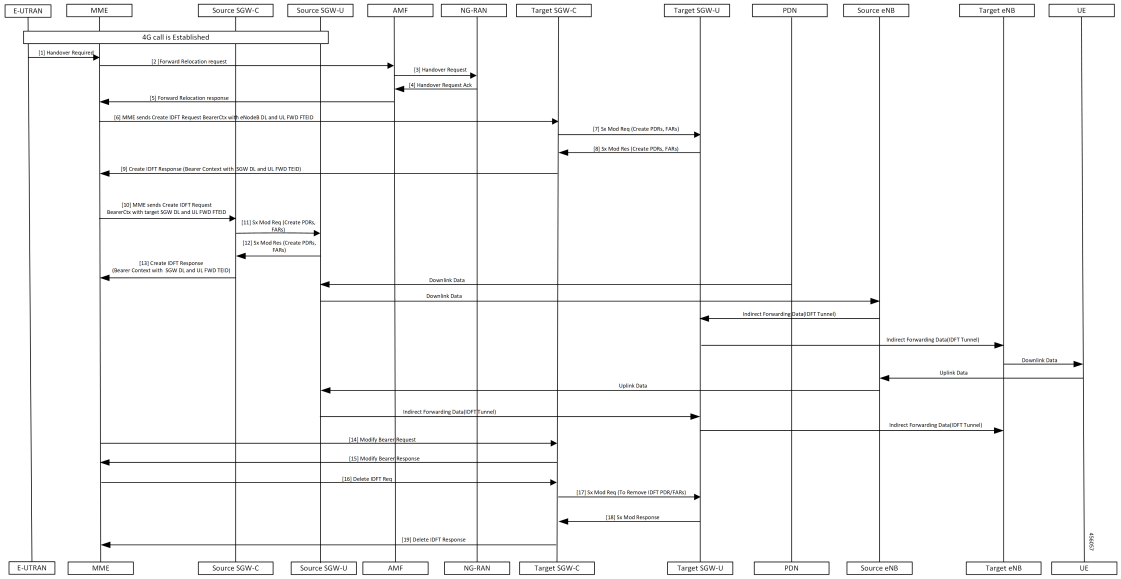


表 *124 : Pure-S* コールの *4G* から *5G* へのハンドオーバーのコールフローの説明

ステップ	説明
1	4G コールが確立されます。 E-UTRAN が MME にハンドオーバーが必要であることを示すメッセージを送信します。
2	MME は転送再配置要求を AMF に送信します。
3	AMF が、NG-RAN にハンドオーバー要求メッセージを送信します。
4	AMF がハンドオーバー要求の ACK メッセージを受信します。
5	MME が AMF から再配置応答を受信します。
6	MME が、eNodeB のダウンリンクおよびアップリンクの転送用 FTEID を含むベアラーコンテキストとともに、IDFT 作成要求をターゲット SGW-C に送信します。
7	ターゲット SGW-C が、PDR および FAR の作成を含む Sx 変更要求をターゲット SGW-U に送信します。
8	ターゲット SGW-U が、PDR および FAR 作成を含む Sx 変更応答をターゲット SGW-C に送信します。

ステップ	説明
9	ターゲット SGW-C が、eNodeB DL と UL FWD FTEID とともに、ベアラーコンテキストとともに IDFT 作成要求を MME に送信します。
10	MME が、ターゲット SGW ダウンリンクおよびアップリンク転送用 FTEID を含むベアラーコンテキストとともに IDFT 作成要求を送信元 SGW-C に送信します。
11	送信元 SGW-C が、PDR および FAR 作成を含む Sx 変更要求を送信元 SGW-U に送信します。
12	送信元 SGW-C が、PDR および FAR 作成を含む Sx 変更応答を送信元 SGW-U から受信します。
13	MME が、SGW ダウンリンクおよびアップリンク転送用 FTEID を含むベアラーコンテキストとともに、IDFT 作成応答を送信元 SGW-C から受信します。 PDN がダウンリンクデータを送信元 SGW-U に送信し、送信元 SGW-U はダウンリンクデータを送信元 eNodeB に送信します。 間接転送データ（IDFT トンネル）は送信元 eNodeB からターゲット SGW-U へ、およびターゲット SGW-U からターゲット eNodeB への方向で始まります。 ターゲット eNodeB が UE にダウンリンクデータを送信します。 UE がアップリンクデータを送信元 eNodeB に送信し、送信元 eNodeB はアップリンクデータを送信元 SGW-U に送信します。 送信元 SGW-U は間接転送データをターゲット SGW-U に送信し、ターゲット SGW-U は間接転送データをターゲット eNodeB に送信します。
14	MME がターゲット SGW-C にベアラー変更要求を送信します。
15	MME がターゲット SGW-C からベアラー変更応答を受信します。
16	MME がターゲット SGW-C に IDFT 削除要求を送信します。
17	ターゲット SGW-C が、ターゲット SGW-U に Sx 変更要求を送信し、PDR と FAR を削除します。
18	ターゲット SGW-C が Sx 変更応答をターゲット SGW-U から受信します。
19	MME がターゲット SGW-C から IDFT 削除応答を受信します。

IDFT（システムレベル）の作成コールフロー

ここでは、IDFT（システムレベル）の作成コールフローについて説明します。

図 *61 : IDFT*（システムレベル）の作成コールフロー

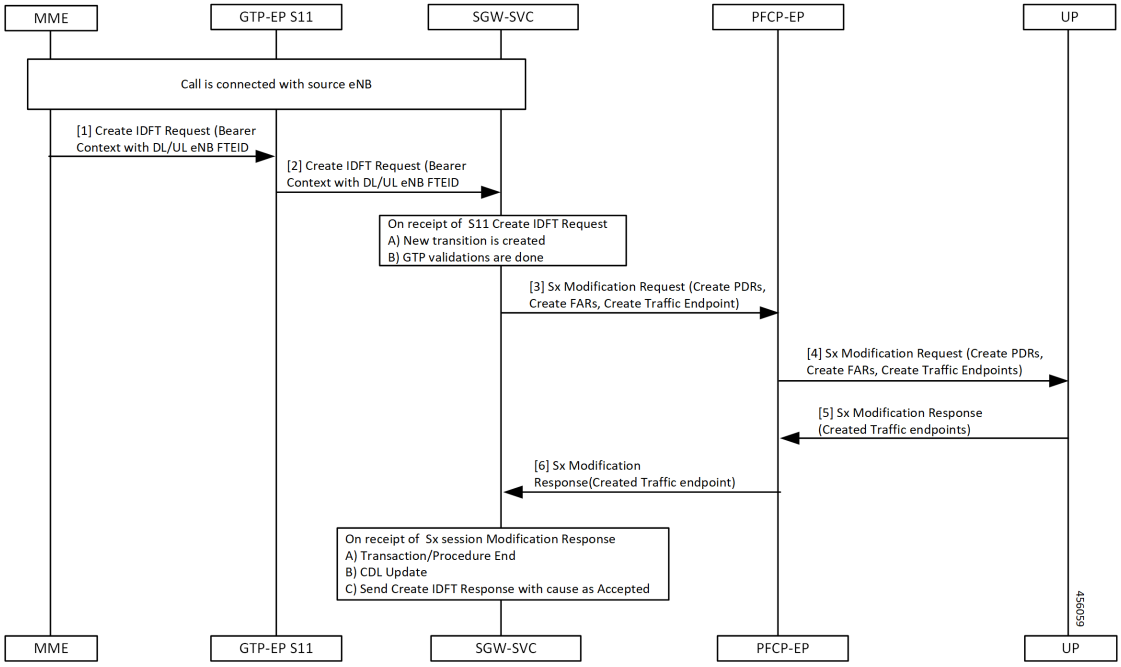


表 *125 : IDFT*（システムレベル）の作成コールフローの説明

ステップ	説明
1	コールが送信元 eNodeB に接続されます。 MME が、DL/UL enB FTEID を持つベアラーコンテキストとともに S11 IDFT 作成要求を GTP-EP S11 に送信します。
2	GTP-EP S11 が、DL/UL enB FTEID を持つベアラーコンテキストとともに S11 IDFT 作成要求を SGW-SVC に送信します。 SGW-SVC が S11 IDFT 作成要求を受信し、次の操作を実行します。 • 新しいトランザクションの作成 • GTP 検証を完了
3	SGW-SVC が、PDR の作成、FAR の作成、およびトラフィックエンドポイントの作成を含む Sx 変更要求を PFCP-EP に送信します。
4	PFCP-EP が、PDR の作成、FAR の作成、およびトラフィックエンドポイントの作成を含む Sx 変更要求を UP に送信します。
5	UPF が、作成されたトラフィックエンドポイントを含む Sx セッション変更応答を PFCP-EP に送信します。

ステップ	説明
6	SGW-SVC は、PFCP-EP から Sx セッション変更応答を受信し、次の操作を実行します。 • トランザクション/手順の終了 • CDL の更新 • Accepted を原因として IDFT 作成応答を送信

IDFT（システムレベル）の削除コールフロー

このセクションでは、IDFT（システムレベル）の削除コールフローについて説明します。

図 *62 : IDFT*（システムレベル）の削除コールフロー

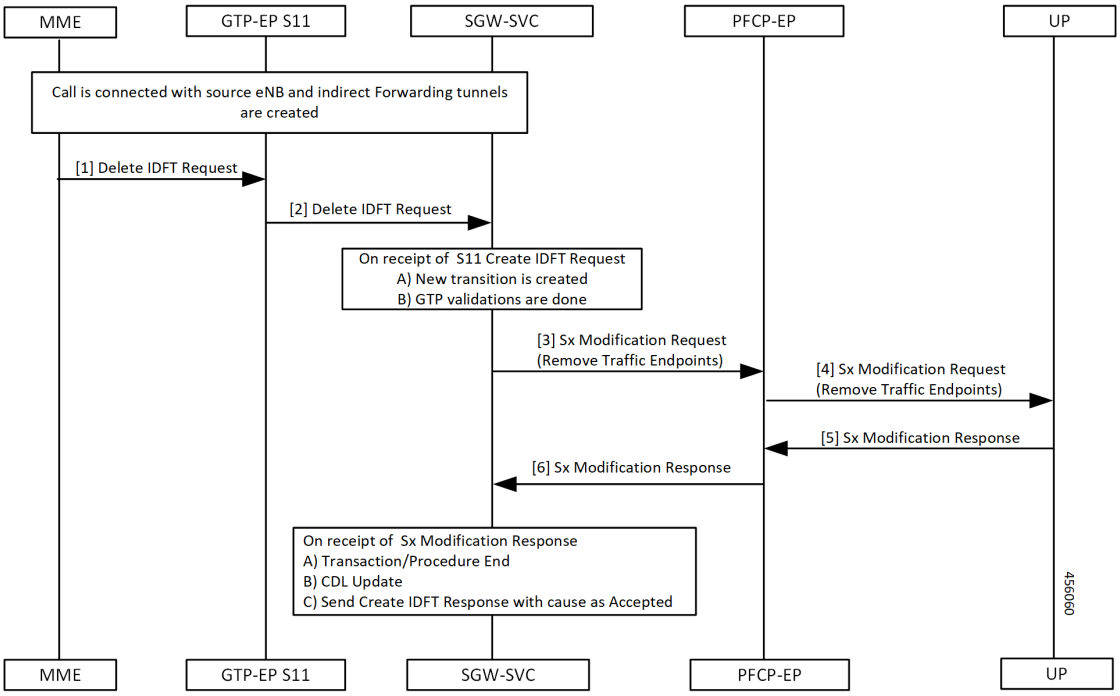


表 *126 : IDFT*（システムレベル）の削除コールフローの説明

ステップ	説明
1	コールが送信元の eNodeB に接続され、間接転送トンネルが作成されます。 MME が、DL/UL enB FTEID を持つベアラーコンテキストとともに S11 IDFT 削除要求を GTP-EP S11 に送信します。

ステップ	説明
2	GTP-EP S11 が、DL/UL enB FTEID を持つベアラーコンテキストとともに S11 IDFT 削除要求を SGW-SVC に転送します。 SGW-SVC が S11 IDFT 削除要求を受信し、次の操作を実行します。 • 新しいトランザクションの作成 • GTP 検証の完了
3	SGW-SVC が、トラフィックエンドポイントの削除を含む Sx 変更要求を PFCP-EP に送信します。
4	PFCP-EP が、トラフィックエンドポイントの削除を含む Sx 変更要求を UP に送信します。
5	UP が Sx セッション変更要求を PFCP-EP に送信します。
6	SGW-SVC は、PFCP-EP から Sx セッション変更応答を受信し、次の操作を実行します。 • トランザクション/手順の終了 • CDL の更新 • 原因を Accepted として、IDFT 削除応答を送信

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

IDFT 構成の表示

ここでは、IDFT 構成を表示するコマンドについて説明します。

UE の非アクティブ IDFT

コマンド： **show subscriber namespace sgw imsi 123456789012345**

出力：

```
{ "subResponses": [ { "status": true,
…
…
"pdnInfoList": { "totalPdn": 1,
"bearerInfoList": {
"totalBearer": 1, "bearerInfo": [ { "bearerId": "Bearer-1", "state": "Connected",
…
…
```

```
} }
]
}
```


1 つのベアラーを持つ 1 つの PDN を持つ UE のアクティブな IDFT

コマンド：**show subscriber namespace sgw imsi 123456789012345**

出力：


```
{ "subResponses": [ { "status": true,
…
…
"pdnInfoList": { "totalPdn": 1,
"bearerInfoList": {
"totalBearer": 1, "bearerInfo": [ { "bearerId": "Bearer-1", "state": "Connected",
…
…
"IndirectForwardingInfo": {
"UplinkInfo":{
"localTeid": "[0x1100000e] 285212686", "localIPv4Address": "209.165.201.8", "remoteTeid":
 "[0x1100000f] 285212687", "remoteIPv4Address": "209.165.201.8",
}
"DownlinkInfo":{
"localTeid": "[0x1100000e] 285212686", "localIPv4Address": "209.165.201.8", "remoteTeid":
 "[0x1100000f] 285212687", "remoteIPv4Address": "209.165.201.8",
}
}
}
]
}
```


ダウンリンク方向に 1 つのベアラーを持つ 1 つの PDN を持つ UE のアクティブ IDFT

コマンド：**show subscriber namespace sgw imsi 123456789012345**

出力：


```
{ "subResponses": [ { "status": true,
…
…
"pdnInfoList": { "totalPdn": 1,
"bearerInfoList": {
"totalBearer": 1, "bearerInfo": [ { "bearerId": "Bearer-1", "state": "Connected",
…
…
"IndirectForwardingInfo": {
"DownlinkInfo":{
"localTeid": "[0x1100000e] 285212686",
"localIPv4Address": "209.165.201.8",
"remoteTeid": "[0x1100000f] 285212687",
"remoteIPv4Address": "209.165.201.8",
}
}
}
]
}
```


1 つの PDN に 2 つのベアラーを持つ UE に対する 1 つのベアラーのアクティブ IDFT

コマンド：**show subscriber namespace sgw imsi 123456789012345**

出力：

```
"subResponses": [ { "status": true,
…
…
"pdnInfoList": { "totalPdn": 1,
"bearerInfoList": {
"totalBearer": 2, "bearerInfo": [ { "bearerId": "Bearer-1", "state": "Connected",
…
…
"IndirectForwardingInfo": {
“UplinkInfo”:{
"localTeid": "[0x1100000e] 285212686", "localIPv4Address": "209.165.201.8", "remoteTeid":
 "[0x1100000f] 285212687", "remoteIPv4Address": "209.165.201.8",
}
“DownlinkInfo”:{
"localTeid": "[0x1100000e] 285212686", "localIPv4Address": "209.165.201.8", "remoteTeid":
 "[0x1100000f] 285212687", "remoteIPv4Address": "209.165.201.8",
}
}
"bearerInfo": [ { "bearerId": "Bearer-2", "state": "Connected",
…
}
}
]
}
```

（注） 表示される IndirectForwardingInfo ブロックは、間接転送トンネルを持つベアラー専用です。

障害処理

cnSGW-C は、IDFT の作成または削除要求手順の障害処理をサポートしています。

メッセージの処理中に発生する可能性のある障害のタイプは次のとおりです。

- 要求と応答の事前検証での失敗
- 再送信タイムアウト
- トランザクション SLA
- 障害は、メッセージ処理の段階に応じて、ピア（UP/PGW/MME）から報告されます。

次の表に、コール処理におけるさまざまな障害シナリオでの cnSGW-C の動作を示します。

障害シナリオ	SGW-SVC の動作	シグナリング（S11）
1. IDFT 作成要求の事前検証の失敗。 2. cnSGW には、IDFT 作成要求で受信したどの EBI でもベアラーコンテキストがありません。	Sx 経由で障害またはシグナリングなしを送信します。	負の IDFT 作成応答。

障害シナリオ	SGW-SVC の動作	シグナリング（S11）
単一の PDN **1.** 単一の PDN を使用した Sx セッション変更要求（IPC、再送信、内部障害など） **2.** 単一の PDN を使用した Sx セッション変更応答（Cause!= ACCEPTED） **3.** Sx 変更応答の検証の失敗	障害を送信します。 存在しない EBI の「コンテキストが見つかりませんでした」を送信します。 sxCause = Context Not Found の場合は PDN をクリアします。	負の IDFT 作成応答。 sxCause = Context Not Found の場合、S11 および S5 経由で DBR および DSR を送信します。
マルチ PDN（部分的エラー） **1.** 部分的な既存の PDN：既存の PDN で続行 **2.** 一部の PDN の Sx セッション変更要求（IPC、再送信タイムアウト、内部障害など） **3.** 一部の PDN の Sx セッション変更応答（Cause!= ACCEPTED） **4.** Sx 変更応答の検証の失敗	存在しない PDN の「コンテキストが見つかりませんでした」を送信します。 Sx 変更要求が失敗した PDN のベアラーコンテキストで障害を送信します。	部分的に承認された IDFT 作成応答。 sxCause = Context Not Found である PDN で、S11 および S5 経由で DBR および DSR を送信します。
マルチ PDN（完全な障害）： **1.** 部分的な既存の PDN：既存の PDN で続行します。 **2.** Sx セッション変更要求（IPC、再送信タイムアウト、内部障害など） **3.** Sx セッション変更応答（Cause!= ACCEPTED） **4.** Sx 変更応答の検証の失敗	存在しない PDN の「コンテキストが見つかりませんでした」を送信します。 Sx 変更要求が失敗した PDN のベアラーコンテキストで障害を送信します。	負の IDFT 作成応答。 sxCause = Context Not Found である PDN で、S11 および S5 経由で DBR および DSR を送信します。
IDFT 削除要求の事前検証の失敗。	Sx 経由で障害またはシグナリングなしを送信します。	負の IDFT 削除応答。

障害シナリオ	SGW-SVC の動作	シグナリング（S11）
1. 単一 PDN とマルチ PDN 2. Sx セッション変更要求（IPC、再送信タイムアウト、内部障害など） 3. Sx セッション変更応答（Cause!= ACCEPTED）	障害を無視します。	正の IDFT 削除応答。 sxCause = Context Not Found である PDN で、S11 および S5 経由で DBR および DSR を送信します。

バルク統計サポート

IDFT サポート機能では、次の統計がサポートされています。

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",
instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
sgw_procedure_type="create_indirect_data_forwarding_tunnel",status="attempted",sub_fail_reason=""}
 3
```

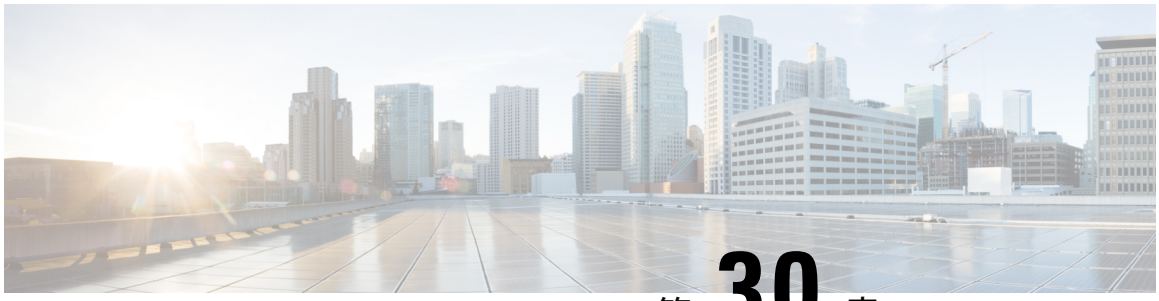
```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",instance_id="0",
interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=
 "create_indirect_data_forwarding_tunnel",status="success",sub_fail_reason=""}
 2
```

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",instance_id="0",
interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=
 "delete_indirect_data_forwarding_tunnel",status="attempted",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",
instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
sgw_procedure_type="delete_indirect_data_forwarding_tunnel",status="success",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",instance_id="0",
interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=
"indirect_data_forwarding_tunnel_guard_timer_expiry",status="attempted",sub_fail_reason=""}
 1
```

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",instance_id="0",

interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",sgw_procedure_type=

"indirect_data_forwarding_tunnel_guard_timer_expiry",status="success",sub_fail_reason=""}
 1
```



第 30 章

アイドル セッション タイムアウトの構成

- 機能の概要と変更履歴 （353 ページ）
- 機能説明 （354 ページ）
- 機能の仕組み （354 ページ）
- 機能設定 （360 ページ）

機能の概要と変更履歴

要約データ

表 *127* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	無効：有効にするには設定が必要です
関連資料	該当なし

マニュアルの変更履歴

表 *128* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

古いセッションタイムアウトにより、SGW-U セッションが終了する前に非アクティブ状態を維持できる期間が決まります。cnSGW-C プラットフォームでは、SubscribeCtx はサブスクライバセッションを表します。SGW-U は、次のようなピアとの接続を確立します。

- S11 または S4 インターフェイスを使用している MME および PGW
- S5 または S8 インターフェイスの PGW

ピアでピアセッションが削除された場合に、SGW-U が削除メッセージを受信しないか、誤ってメッセージを見逃すような状況では、SGW-Uセッションはアイドル状態のままになり、コールは受信し続けますが、処理したり、要求に応答したりはしません。古いセッションでリソースが使用されないようにするために、アイドルタイムアウト機能を使用すると、古いセッションの削除後に新しいサブスクライバセッション要求を SGW-U が受信できます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

サブスクライバセッションは、一定時間データトラフィックのアクティビティがなく、そのセッションに向けられないと、アイドル状態となります。

ユーザープレーンのセッションマネージャは、コール回線の状態を追跡します。セッションマネージャがコール回線データトラフィックを記録していないセッションは、アイドル状態と判断されます。アイドル セッション タイムアウト設定を使用すると、セッションがタイムアウトする前にアイドル状態を維持できる時間間隔を設定できます。アイドルタイムアウト設定は、セッションが確立されたときに設定されます。SGW-U は、Sx セッション確立要求でタイムアウト設定をユーザープレーンに送信します。マルチ PDN コールの場合、古いセッションに向けられたコールは、すべての PDN に対して非アクティブレポートが生成された後にクリアされます。

毎秒、SGW-U はデータ トラフィック アクティビティをモニタリングして、セッションのアイドル状態を判断します。古いセッションを特定すると、ユーザープレーンは Sx セッション使用状況レポートのユーザープレーン非アクティブレポートを更新し、cnSGW-C に送信して、セッションがアイドル状態であることを伝えます。さらに、cnSGW-C はピアへのセッション削除要求を開始します。

ネットワーク環境に基づいて、アイドルタイムアウト設定を秒単位で設定します。許可されるタイムアウト値の範囲は 1 ～ 4294967295 秒です。タイムアウトの設定は、SGW-U サービスプロファイルレベルで適用され、SGW-U サービスによって処理される一連のサブスクライバのアイドルタイムアウト処理を有効にします。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

非アクティブ レポート コールフロー

ここでは、非アクティブ レポート コールフローについて説明します。

図 *63 :* 非アクティブ レポート コールフロー

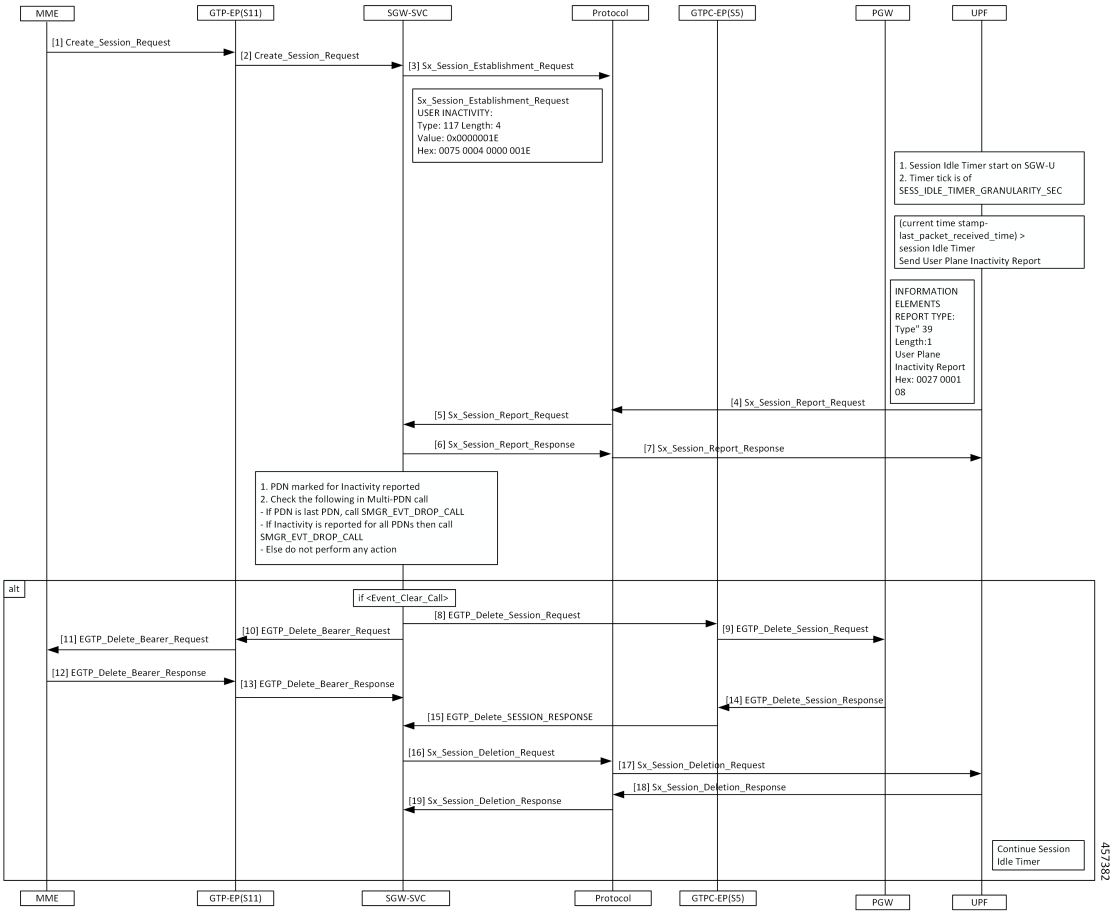


表 *129 :* 非アクティブ レポート コールフローの説明

ステップ	説明
1	MME が Create Session Request を GTP-EP（S11）に送信します。
2	GTP-EP（S11）が Create Session Request を SGW に転送します。
3	SGW-SVC がプロトコルへの Sx Session Establishment Request の一部として、IE＝USER INACTIVITY でセッションアイドルタイマーを送信します。

ステップ	説明
4	SGW-U が USER INACTIVITY からのセッションアイドルタイマーを読み取り、CLP レベルで保存します。 UPF が Sx Session Report Request をプロトコルに送信します。
5	プロトコルが Sx Session Report Request を SGW-SVC に送信します。
6	SGW-SVC が Sx Session Report Response をプロトコルに送信します。
7	プロトコルが Sx Session Report Response を UPF に送信します。
8	SGW-SVC が Delete Session Request を GTPC-EP に送信します。
9	GTPC-EP が Delete Session Request を PGW に送信します。
10	SGW-SVC が Delete Bearer Request を GTP-EP に送信します。
11	GTP-EP が Delete Bearer Request を MME に送信します。
12	MME が Delete Bearer Response を GTP-EP に送信します。
13	GTP-EP が Delete Bearer Response を SGW-SVC に送信します。
14	PGW が Delete Session Response を GTPC-EP に送信します。
15	GTPC-EP が EGTP Delete Session Response を SGW-SVC に送信します。
16	SGW-SVC が Sx Session Deletion Request をプロトコルに送信します。
17	プロトコルが Sx Session Deletion Request を UPF に送信します。
18	UPF が Sx Delete Session Response をプロトコルに送信します。
19	プロトコルが Sx Delete Session Response を SGW-SVC に送信します。

UPF におけるアイドルタイマーの処理コールフロー

ここでは、UPF の Create Session Request でアイドルタイマーを受信した場合コールフローについて説明します。

図 *64 : UPF* におけるアイドルタイマーの処理コールフロー

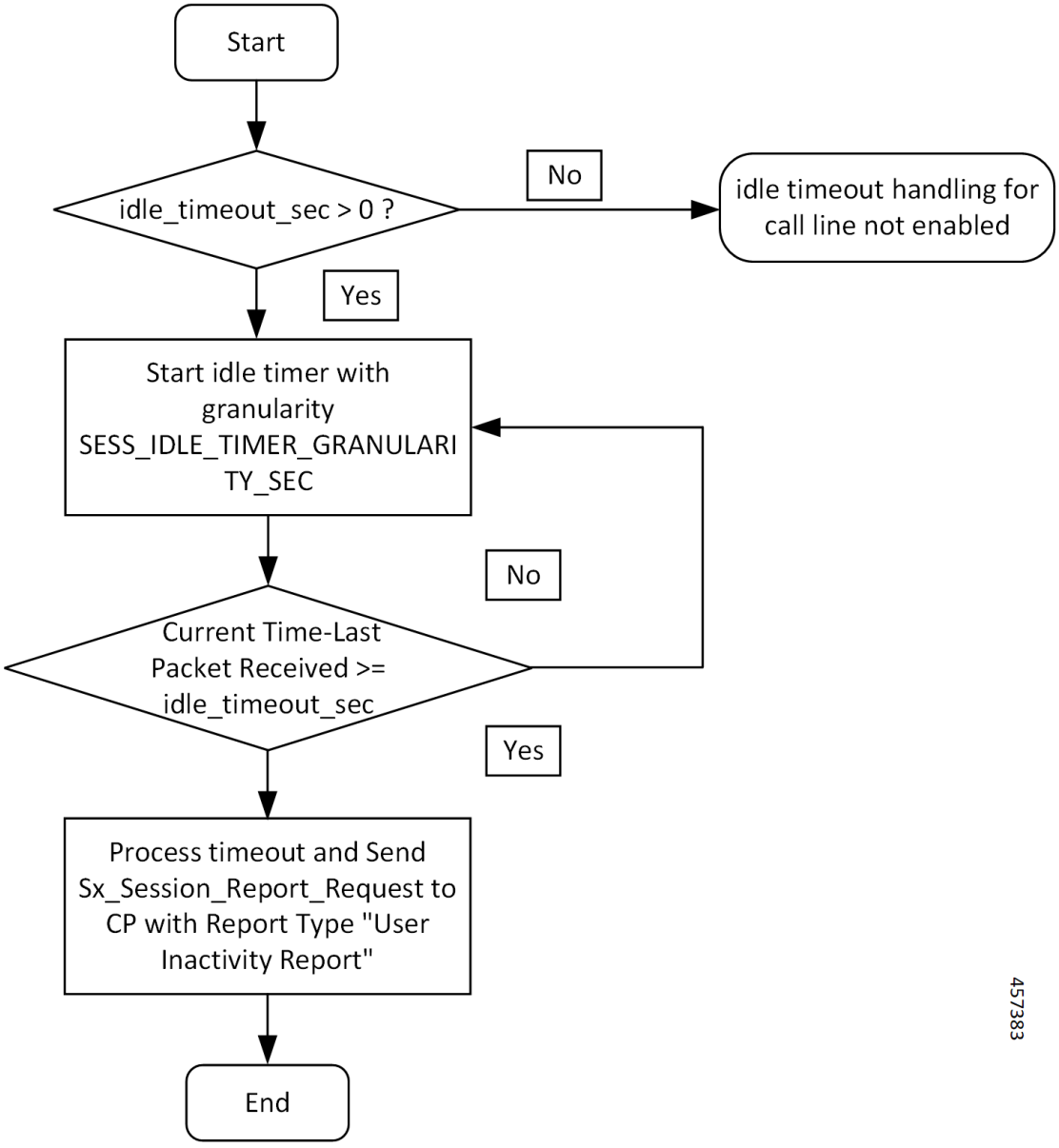


表 *130 : UPF* におけるアイドルタイマーの処理コールフローの説明

ステップ	説明
1	idle_timeout_sec の値が 0 より大きい場合、タイマーは 1 秒の粒度で UPF で開始されます。 それ以外の場合、アイドルタイムアウトが無効になります。

ステップ	説明
2	タイマーは毎秒タイムアウトします。現在の時刻と最後のパケット受信時刻の差が idle_timeout_sec よりも大きいかどうかを確認します。時間の差が idle_timeout_sec よりも大きい場合、UP が、レポートタイプ=UPIR（非アクティブレポート）の Sx Session Report Request を CP（cnSGW-C）に送信します。

再アクティブ レポート コール フロー

ここでは、再アクティブ レポート コール フローについて説明します。

図 *65 :* 再アクティブ レポート コール フロー

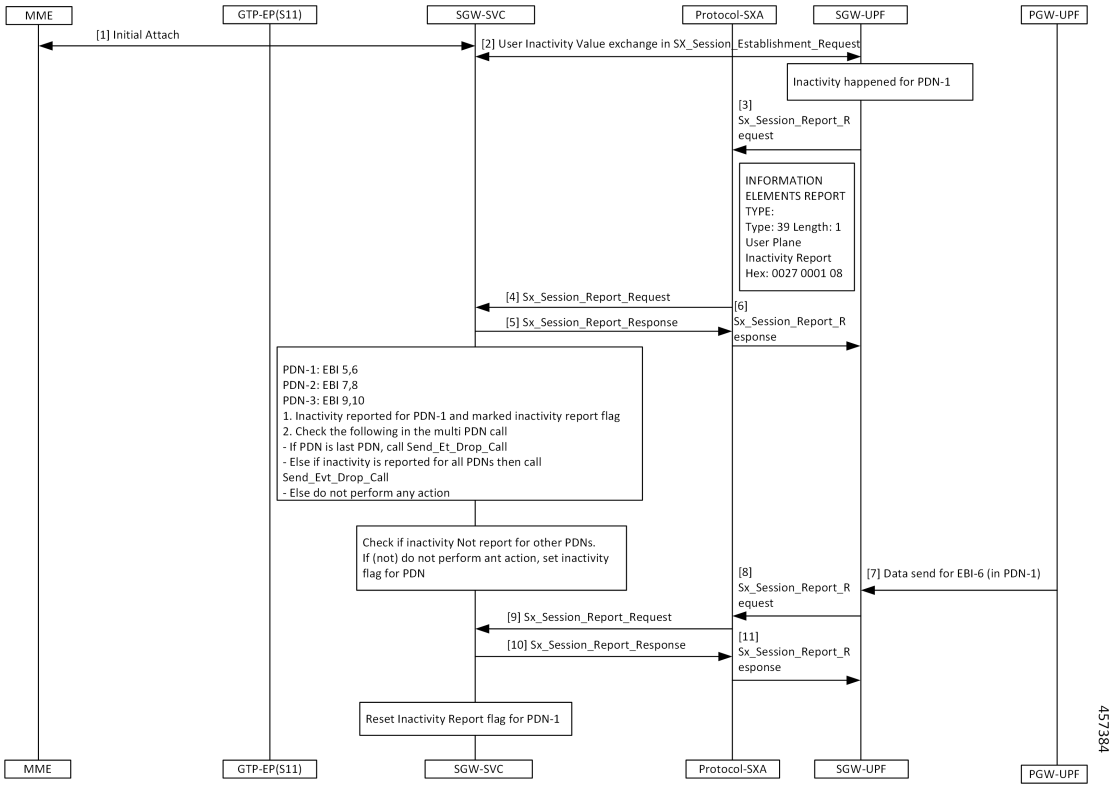


表 *131 :* 再アクティブ レポート コール フローの説明

ステップ	説明
1	MME と SGW-SVC が最初のアタッチ要求を処理します。

ステップ	説明
2	Protocol-SXA と SGW-UPF が、Sx セッション確立要求でユーザーの非アクティブ情報の交換を実行します。
3	PDN-1 で非アクティブが確認された場合、SGW-UPF は、タイプがユーザープレーン非アクティブレポートである Sx_Session_Report_Request をプロトコル SXA に送信します。
4	Protocol-SXA が Sx_Session_Report_Request を SGW-SVC に送信します。
5	SGW-SVC が Sx_Session_Report_Response を Protocol-SXA に送信します。
6	Protocol-SXA が Sx_Session_Report_Response を SGW-UPF に転送します。
7	PGW-UPF が EBI-6（PDN-1 内）のデータを SGW-UPF に送信します。
8	SGW-UPF が、IE レポートタイプにユーザープレーン再アクティブレポートを使用して Sx_Session_Report_Request を Protocol-SXA に送信します。
9	Protocol-SXA が、IE レポートタイプにユーザープレーン再アクティブレポートを使用して Sx_Session_Report_Request を SGW-SVC に送信します。
10	SGW-SVC が Sx_Session_Report_Response で Protocol-SXA に応答します。
11	Protocol-SXA が Sx_Session_Report_Response を SGW-UPF に送信します。 コントロールプレーンで Sx_Session_Report_Response を受信した後、SGW-SVC は PDN の非アクティブレポートフラグをクリアします。

ClearCall 処理コールフロー

ここでは、ClearCall 処理コールフローについて説明します。

図 *66 : ClearCall* 処理コールフロー

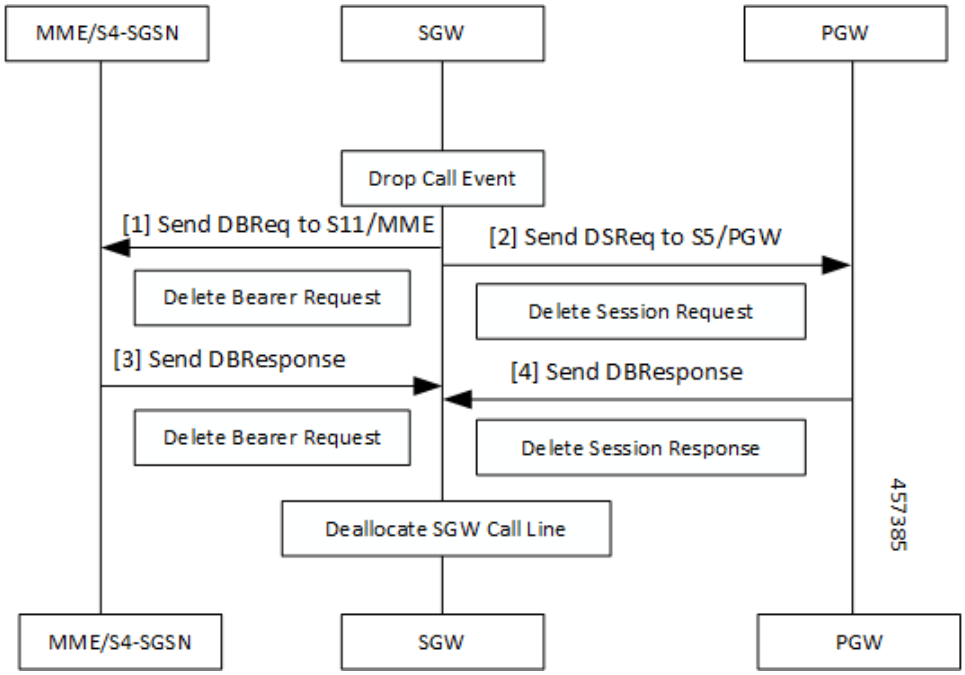


表 *132 : ClearCall* 処理コールフローの説明

ステップ	説明
1	SGW で、コントロールプレーンが Sx Session Report Request でユーザープレーン非アクティブレポートを受信します。コントロールプレーンが、要求の PDN が最新の PDN であるか、または非アクティブレポートですでに他の PDN が報告されているかを評価し、ClearCall の手順を開始します。
2	ClearCall メッセージを受信すると、cnSGW-C がピアへの Session Deletion Request をトリガーします。
3	SGW が Delete Bearer Request を S11/MME に送信します。
4	SGW が Delete Session Request を S5/PGW に送信します。
5	Delete Session Request を受信すると、SGW が Sx Delete Session Request を送信して、UPF 上のリソースをクリアします。

機能設定

この機能を設定するには、次の設定を使用します。

```
config
  profile sgw sgw_group_name
    session-idle-timer session_idle_timer
    end
```

注：

- **session-idle-timer** *session_idle_timer*：セッションがアイドル状態を維持する最大時間を秒単位で指定します。構成された時間に達すると、システムは自動的にセッションを終了します。許容範囲には、1 ー 4294967295 の整数が含まれます。デフォルト値はゼロで、アイドル セッションが無効であることを示します。

設定例

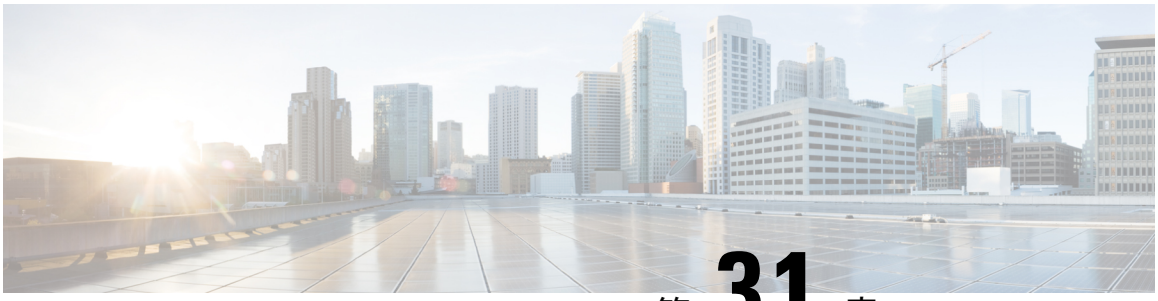
次に、設定例を示します。

```
config
   profile sgw sgw1
     session-idle-timer 1000
     end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw sgw1 session-idle-timer
profile sgw sgw1
session-idle-timer 1000
```



第 31 章

初期接続のサポート

- 機能の概要と変更履歴 （363 ページ）
- 機能説明 （364 ページ）
- 機能の仕組み （364 ページ）
- バックオフタイマー、開始タイムスタンプ、および最大待機時間のサポート （367ページ）

機能の概要と変更履歴

要約データ

表 *133* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *134* : マニュアルの変更履歴

改訂の詳細	リリース
情報要素（IE）のサポートが追加されました。 • バックオフ タイマー • 発信タイムスタンプ • 最長待機時間	2021.02.3

改訂の詳細	リリース
最初の導入。	2020.04

機能説明

cnSGW-C は、初期接続の Create Session Request の処理をサポートしています。この機能の一部として、cnSGW-C は EGTP エンドポイントを介した MME からの Create Session Request の受信をサポートします。さらに、UDP メッセージを復号化し、内部メッセージ処理のためにメッセージを gRPC メッセージに変換します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

初期接続コールフロー

このセクションでは、初期接続コールフローについて説明します。

図 *67* : 初期接続コールフロー

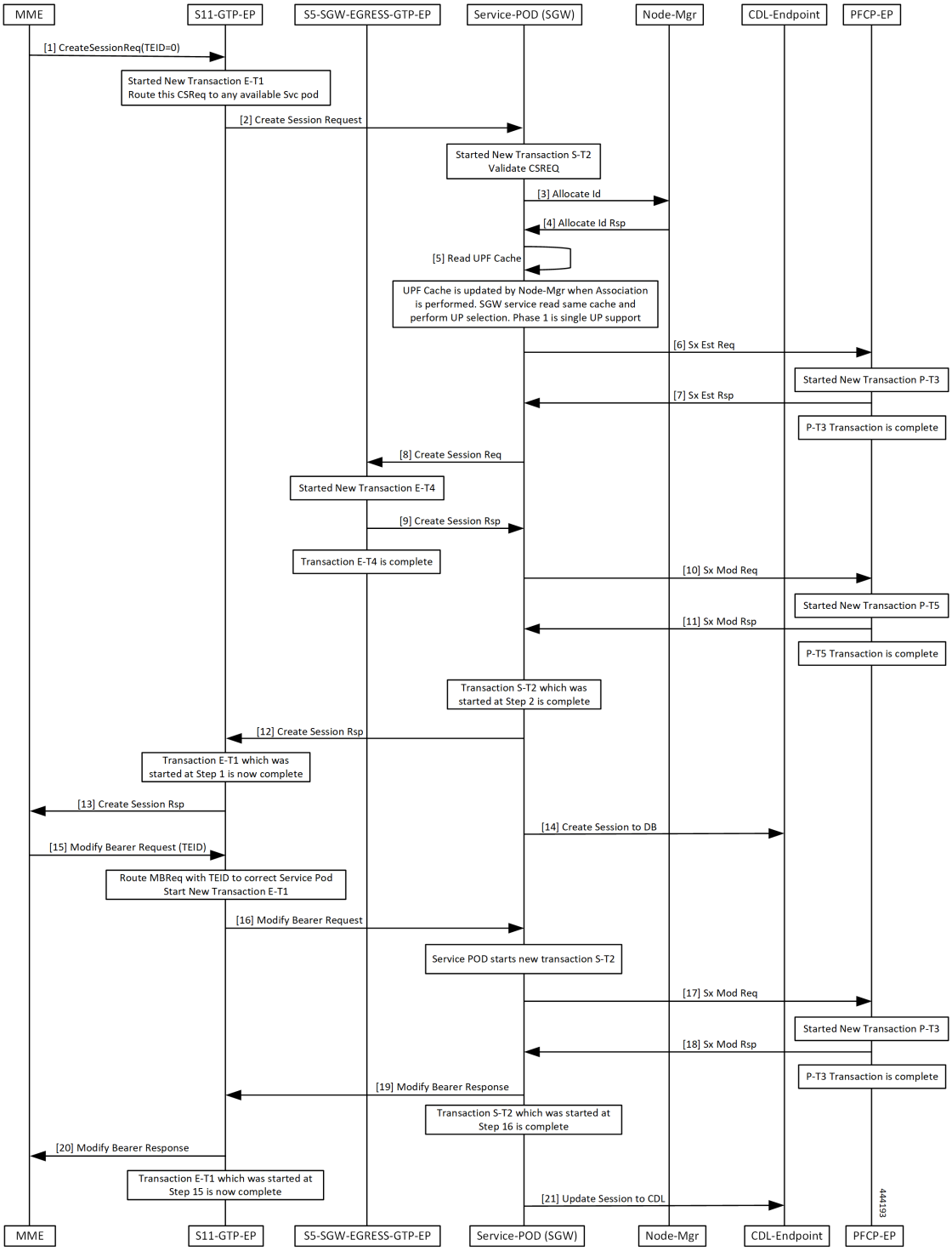


表 *135 :* 初期接続コールフローの説明

ステップ	説明
1	MME が、TEID の値をゼロに設定したセッション作成要求を S11-GTP-EP に送信します。
2	トランザクション E-T1 が開始されます。 S11-GTP-EP がセッション作成要求をサービスポッド（SGW）に転送します。
3	セッション作成要求を検証するためにトランザクション S-T2 が開始されます。 サービスポッド（SGW）が識別子割り当て要求を Node-Mgr に送信します。
4	Node-Mgr が、識別し割り当て応答をサービスポッド（SGW）に送信します。
5	サービスポッド（SGW）がキャッシュを読み取り、UPF 選択を実行します。
6	サービスポッド（SGW）が Sx 確立要求を PFCP-EP に送信します。
7	トランザクション P-T3 が開始されます。 PFCP-EP は、Sx 確立応答をサービスポッド（SGW）に送信します。
8	トランザクション P-T3 が完了しました。 サービスポッド（SGW）がセッション作成要求を S5-SGW-EGRESS-GTP-EP に送信します。
9	トランザクション E-T4 が開始されます。 S5-SGW-EGRESS-GTP-EP が、セッション作成応答をサービス POD（SGW）に送信します。
10	トランザクション E-T4 が完了しました。 サービスポッド（SGW）が Sx 変更要求を PFCP-EP に送信します。
11	トランザクション P-T5 が開始されます。 PFCP-EP が、Sx 変更応答をサービスポッド（SGW）に送信します。
12	トランザクション P-T5 と S-T2 が完了しました。 サービスポッド（SGW）が、セッション作成応答を S11-GPT-EP に送信します。
13	トランザクション E-T1 が完了しました。 S11-GPT-EP が、セッション作成応答を MME に転送します。
14	サービスポッド（SGW）が、DB へのセッション作成メッセージを CDL エンドポイントに送信します。

ステップ	説明
15	MME が TEID を使用してベアラー変更要求を S11-GTP-EP に送信します。
16	トランザクション E-T1 が開始されます。 S11-GTP-EP がベアラー変更リクエストをサービスポッド（SGW）に送信します。
17	トランザクション S-T2 が開始されました。 サービスポッド（SGW）が Sx 変更要求を PFCP-EP に送信します。
18	トランザクション P-T3 が開始されます。 PFCP-EP が、Sx 変更応答をサービスポッド（SGW）に送信します。
19	トランザクション P-T3 が完了しました。 サービスポッド（SGW）が、ベアラー変更応答を S11-GTP-EP に送信します。
20	トランザクション S-T2 が完了しました。 S11-GPT-EP が、ベアラー変更応答を MME に転送します。
21	トランザクション E-T1 が完了しました。 サービスポッド（SGW）が、CDL へのセッション更新メッセージを CDL-Endpoint に送信します。セッションが CDL で更新されます。

標準準拠

この機能は、次の標準仕様に準拠しています。

- *3GPP TS 29.274 "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3"*

バックオフタイマー、開始タイムスタンプ、および最大待機時間のサポート

このセクションでは、情報要素（IE）であるバックオフタイマー、開始タイムスタンプ、および最大待機時間のサポートについて説明します。

IE の技術仕様の詳細については、*3GPP TS 29.274* を参照してください。

バックオフ タイマー

バックオフ時間は、サービスユーザーと緊急サービス以外のサービスのために、MME または S4-SGSN が、輻輳した APN の PGW に後続の PDN 接続確立要求を送信することを控える時間を示します。

バックオフタイマー機能を使用すると、IPアドレスが使い果たされた場合の障害状態に対する新しい接続を拒否できます。

IP アドレスが使い果たされると、PGW-C/SMF はステータスを失敗として検出し、セッション作成応答にバックオフタイマーを追加します。SGW は、セッション作成応答で MME にバックオフタイマー値を転送します。

発信タイムスタンプ

発信タイムスタンプは、発信エンティティが要求を開始した時刻です。タイムスタンプはUTC形式で表示されます。

MME/SGSN および TWAN/ePDG には、それぞれ S11/S4 および S2a/S2b インターフェイスの発信タイムスタンプ IE が含まれています。

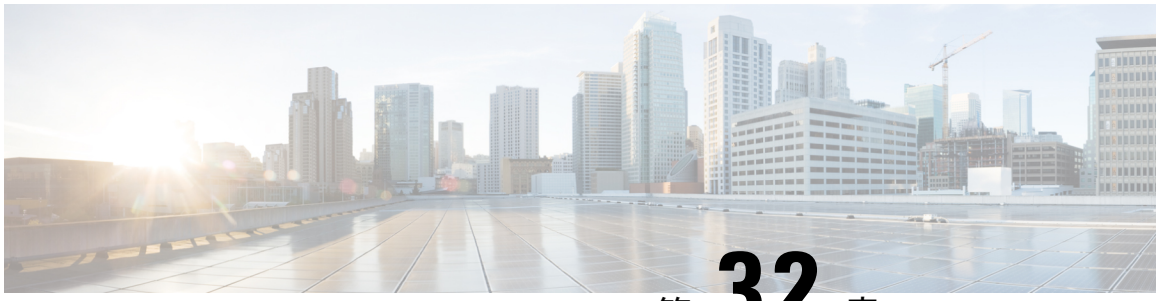
SGW は MME/SGSN から発信タイムスタンプ IE を受信し、S5/S8 インターフェイスの IE を含めます。

MaxWaitTime

MaxWaitTimeは、要求の発信者が応答を待機する期間（発信タイムスタンプが切れてからのミリ秒数）を示します。

MME/SGSN および TWAN/ePDG には、それぞれ S11/S4 および S2a/S2b インターフェイス上の MaxWaitTime IE が含まれています。

SGW には、S5/S8 インターフェイスの MaxWaitTime IE が含まれています。



第 32 章

システム間 RAT ハンドオーバー

- 機能の概要と変更履歴 （369 ページ）
- 機能説明 （370 ページ）
- 機能の仕組み （370 ページ）

機能の概要と変更履歴

要約データ

表 *136* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *137* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

cnSGW-C は、コンバージド コア ネットワーク（4G-5GC）のコントロール プレーン ネットワーク機能（NF）です。

cnSGW-C NF は、SMI アーキテクチャ上に構築されています。cnSGW-C は UE アンカーとして機能し、モビリティ手順に加えて、3GPP TS 23.401、23.214 で指定されているセッションセットアップおよび終了手順をサポートします。

cnSGW-C ユーザープレーン（UP）は、データトラフィックを伝送する UP セッションとベアラーを作成するために使用されます。

この機能は、cnSGW-C の次の手順をサポートします。

- Wi-Fi から LTE
- GnGp から LTE へのハンドオーバー

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

Wi-Fi から LTE への成功コールフロー

ここでは、Wi-Fi から LTE への成功コールフローについて説明します。

図 *68 : Wi-Fi* から *LTE* への成功コールフロー

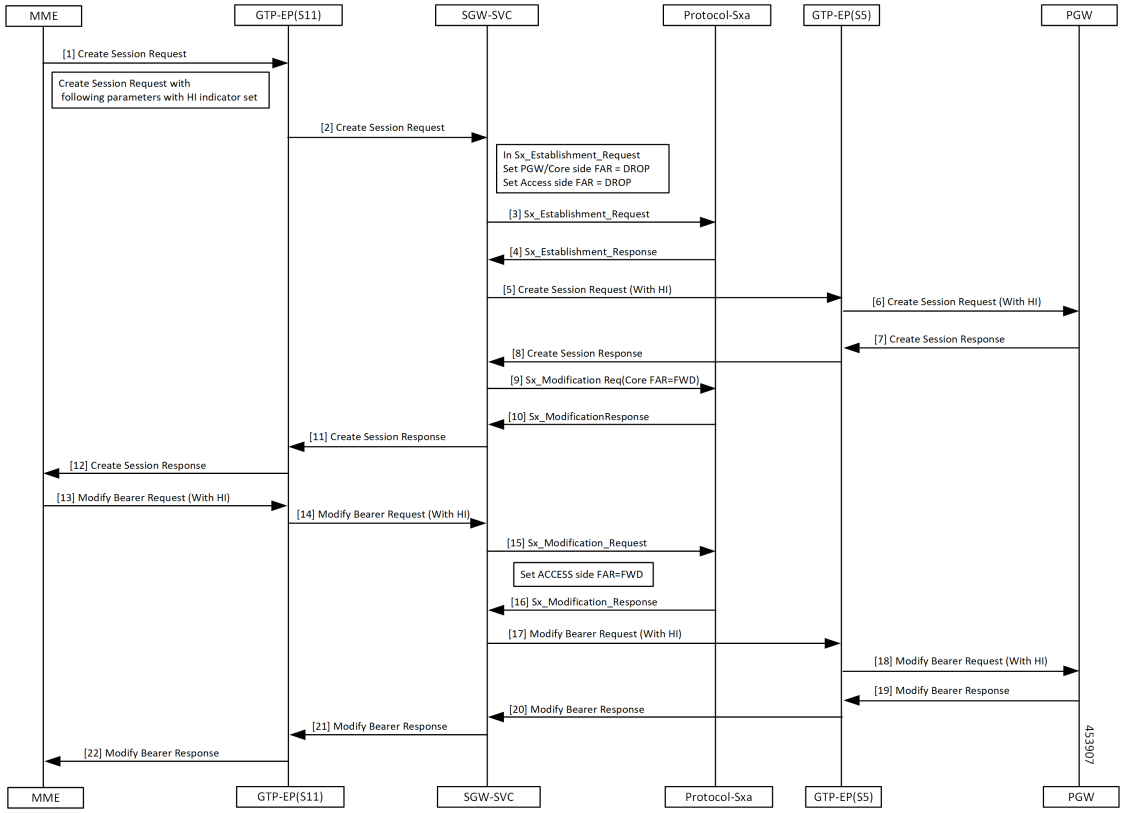


表 *138 : Wi-Fi* から *LTE* への成功コールフローの説明

ステップ	説明
1	MME が、次の情報とともに GTP-EP（S11）にセッション作成要求を送信します。 • EUTRAN としての RAT • TRUE に設定されたハンドオフインジケータ。
2	GTP-EP（S11）がセッション作成要求を SGW-SVC に転送します。
3	SGW-SVC が Sx 確立要求を Protocol-Sxa に送信します。
4	Protocol-Sxa が Sx 確立応答を SGW-SVC に送信します。
5	SGW-SVC がセッション作成要求（HI を使用）を GTP-EP（S5）に送信します。
6	GTP-EP（S5）がセッション作成要求（HI を使用）を PGW に転送します。
7	PGW がセッション作成応答を GTP-EP（S5）に送信します。 PGW が IPv6 プレフィックスを提供します。

ステップ	説明
8	GTP-EP（S5）がセッション作成応答を SGW-SVC に転送します。
9	SGW-SVC が Sx 変更要求を Protocol-Sxa に送信します。
10	Protocol-Sxa が Sx 変更応答を SGW-SVC に送信します。
11	SGW-SVC がセッション作成応答を GTP-EP（S11）に送信します。
12	GTP-EP（S11）がセッション作成応答を MME に送信します。
13	MME がベアラー変更要求（HI を使用）を GTP-EP（S11）に送信します。
14	GTP-EP がベアラー変更要求（HI を使用）を SGW-SVC に転送します。
15	SGW-SVC が Sx 変更要求を Protocol-Sxa に送信します。
16	Protocol-Sxa が Sx 変更応答を SGW-SVC に送信します。
17	SGW-SVC がベアラー変更要求（HI を使用）を GTP-EP（S5）に転送します。
18	GTP-EP（S5）がベアラー変更要求（HI を使用）を PGW に転送します。
19	PGW が GTP-EP（S5）にベアラー変更応答を送信します。
20	GTP-EP（S5）がベアラー変更応答を SGW-SVC に転送します。
21	SGW-SVC がベアラー変更応答を GTP-EP（S11）に転送します。
22	GTP-EP（S11）がベアラー変更応答を MME に転送します。 S1 SGW FTEID は、セッション作成応答で SGW-SVC から MME に送信される S1-U SGW FTEID と同じです。 SGW-SVC はダウンリンクパケットを eNodeB に送信できるようになり、Wi-Fi から LTE へのデータパスの切り替えは、ベアラー変更応答の後に発生します。

OI インジケータの設定を使用した GnGp から LTE へのハンドオーバーコールフロー

ここでは、操作表示（OI）インジケータの設定を使用したGnGpからLTEへのハンドオーバーコールフローについて説明します。

図 *69 : OI* インジケータの設定を使用した *GnGp* から *LTE* へのハンドオーバーコールフロー

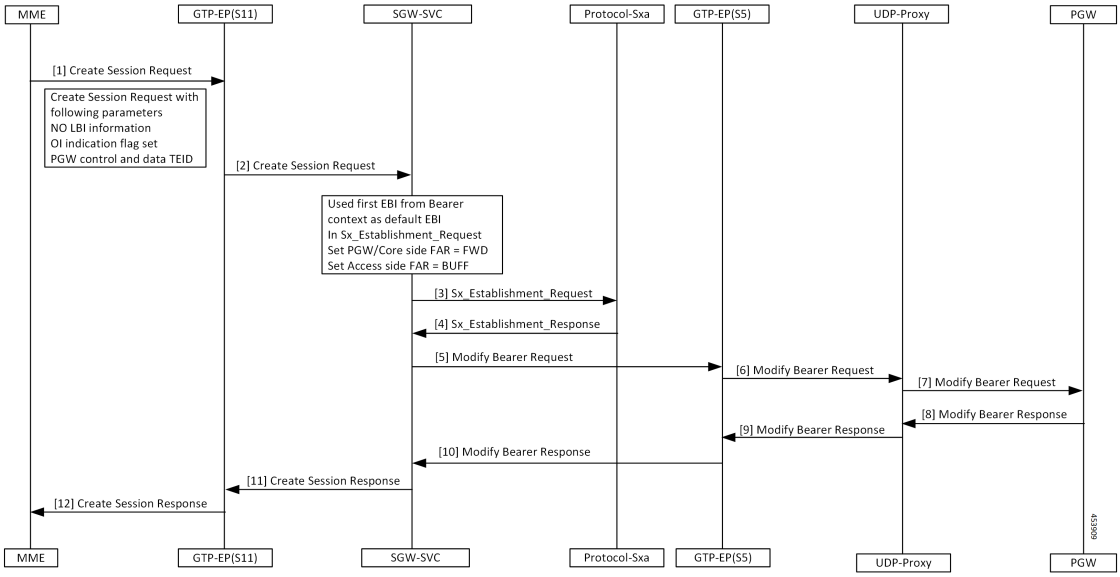


表 *139 : OI* インジケータの設定を使用した *GnGp* から *LTE* へのハンドオーバーコールフローの説明

ステップ	説明
1	MME が次の情報とともに Create Session Request を GTP-EP（S11）に送信します。 • EBI リスト（LBI 情報なし） • PGW 制御およびデータ TEID • OI インジケータフラグセット
2	GTP-EP（S11）が Create Session Request を SGW-SVC に転送します。
3	SGW-SVC が Sx Session Establishment Request を Protocol-Sxa に送信します。
4	Protocol-Sxa が Sx Establishment Response を SGW-SVC に送信します。
5	SGW-SVC が Modify Bearer Request を GTP-EP（S5）に送信します。
6	GTP-EP（S5）が Modify Bearer Request を UDP プロキシに転送します。
7	UDP プロキシが Modify Bearer Request を PGW に転送します。
8	PGW がデフォルトの EBI 情報を含む Modify Bearer Response を UDP プロキシに送信します。
9	UDP プロキシが Modify Bearer Response を GTP-EP（S5）に転送します。
10	GTP-EP（S5）が Modify Bearer Response を SGW-SVC に転送します。

ステップ	説明
11	SGW-SVC がデフォルトの EBI 情報を含む Create Session Response を GTP-EP（S11）に送信します。
12	GTP-EP（S11）が Create Session Response を MME に転送します。

OI インジケータの設定解除を使用した GnGp から LTE へのハンドオーバーコールフロー

ここでは、操作表示（OI）インジケータの設定解除を使用した GnGp から LTE へのハンドオーバーコールフローについて説明します。

図 *70 : OI* インジケータの設定解除を使用した *GnGp* から *LTE* へのハンドオーバーのコールフロー

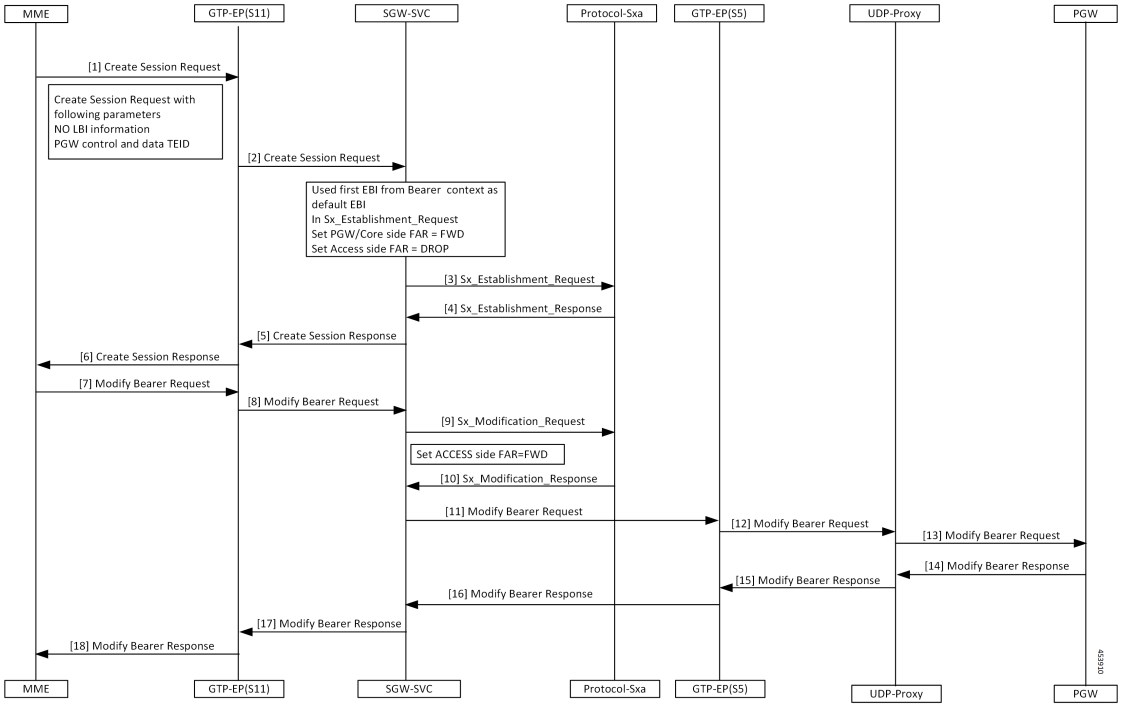


表 *140 : OI* インジケータの設定解除を使用した *GnGp* から *LTE* へのハンドオーバーコールフローの説明

ステップ	説明
1	MME が次の情報とともに Create Session Request を GTP-EP（S11）に送信します。 • EBI リスト（LBI 情報なし） • PGW 制御およびデータ TEID • OI インジケータフラグの設定解除

ステップ	説明
2	GTP-EP（S11）が Create Session Request を SGW-SVC に転送します。
3	SGW-SVC が Sx Session Establishment Request を Protocol-Sxa に送信します。
4	Protocol-Sxa が Sx Establishment Response を SGW-SVC に送信します。
5	SGW-SVC が Create Session Response を GTP-EP（S11）に送信します。
6	GTP-EP（S11）が Create Session Response を MME に転送します。
7	MME が Modify Bearer Request を GTP-EP（S11）に送信します。
8	GTP-EP（S11）が Modify Bearer Request を SGW-SVC に転送します。
9	SGW-SVC が Sx Modification Request を Protocol-Sxa に送信します。
10	Protocol-Sxa が Sx Modification Response を SGW-SVC に送信します。
11	SGW-SVC が Modify Bearer Request を GTP-EP（S5）に送信します。
12	GTP-EP（S5）が Modify Bearer Request を UDP プロキシに転送します。
13	UDP プロキシが Modify Bearer Request を PGW に転送します。
14	PGW がデフォルトの EBI 情報を含む Modify Bearer Response を UDP プロキシに送信します。
15	UDP プロキシが Modify Bearer Response を GTP-EP（S5）に転送します。
16	GTP-EP（S5）が Modify Bearer Response を SGW-SVC に転送します。
17	SGW-SVC が Modify Bearer Response を GTP-EP（S11）に転送します。
18	GTP-EP（S11）が Modify Bearer Response を MME に転送します。 S1 SGW FTEID は、Create Session Response で SGW-SVC から MME に送信される S1-U SGW FTEID と同じです。 SGW-SVC はダウンリンクパケットを eNodeB に送信できるようになり、Wi-Fi から LTE へのデータパスの切り替えは、Modify Bearer Response の後に発生します。

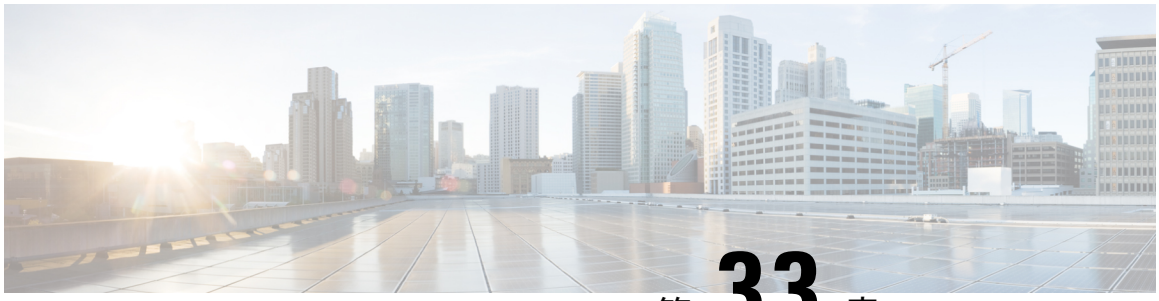
（注） 次のシナリオでは、cnSGW-C は、Modify Bearer Response で受信したデフォルト EBI が最初の EBI と異なる場合にコールをクリアします。

- OI インジケータの設定を使用した GnGp から LTE へのハンドオーバー
- OI インジケータの設定解除を使用した GnGp から LTE へのハンドオーバー

標準準拠

この機能は、次の標準仕様に準拠しています。

- 3GPP TS 23.401「General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access」
- 3GPP TS 23.214「Architecture enhancements for control and user plane separation of EPC nodes」



第 33 章

MME内およびMME間ハンドオーバー手順

- 機能の概要と変更履歴 （377 ページ）
- 機能説明 （378 ページ）
- 機能の仕組み （378 ページ）
- MME 内および MME 間ハンドオーバーの手順の OAM サポート （387 ページ）

機能の概要と変更履歴

要約データ

表 *141* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *142* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

cnSGW-C は、MME 内 SGW 内、および MME 間 SGW 内ハンドオーバーをサポートします。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

MME 間ハンドオーバーのアクティブ-アクティブ移行コールフロー

ここでは、MME 間ハンドオーバーのアクティブ-アクティブ移行コールフローについて説明します。

図 *71 : MME* 間ハンドオーバーのアクティブ-アクティブ移行コールフロー

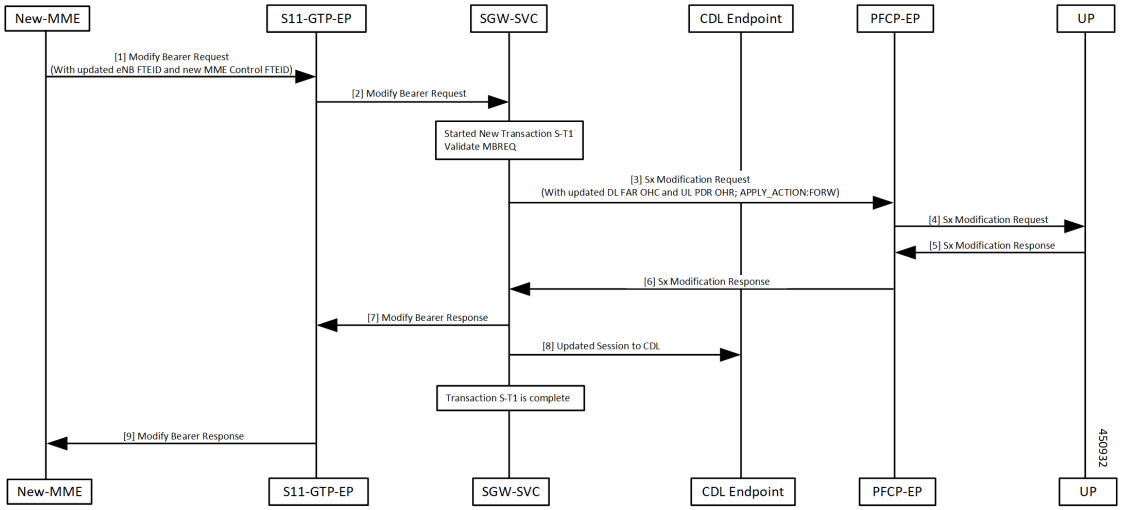


表 *143 : MME* 間ハンドオーバーのアクティブ-アクティブ移行コールフローの説明

ステップ	説明
1	New-MME が、更新された eNodeB F-TEID と新しい MME 制御 F-TEID とともに、S11-GTP-EP ポッドにベアラー変更要求を送信します。

ステップ	説明
2	S11-GTP-EP ポッドが SGW-SVC にベアラー変更要求を転送します。 SGW-SVC は次を実行します。 • 新しいトランザクション S-T1 を作成 • ベアラー変更要求を検証
3	SGW-SVC が、ダウンリンク FAR OHC、アップリンク PDR OHR、および Apply Action を FORWARD として、Sx 変更要求を PFCP-EP ポッドに送信します。
4	PFCP-EP ポッドが Sx 変更要求を UP に送信します。
5	UP が Sx 変更応答を PFCP-EP ポッドに送信します。
6	PFCP-EP ポッドが SGW-SVC に Sx 変更応答を送信します。
7	SGW サービスポッドが S11-GTP-EP ポッドにベアラー変更応答を送信します。
8	SGW-SVC が更新されたセッションを CDL エンドポイントに送信します。 トランザクション S-T1 が完了しました。
9	S11-GTP-EP ポッドが New-MME にベアラー変更応答を送信します。

MME 内ハンドオーバーのアクティブ-アクティブ移行コールフロー

このセクションでは、MME 内ハンドオーバーのアクティブ-アクティブ移行コールフローについて説明します。

図 *72 : MME* 内ハンドオーバーのアクティブ-アクティブ移行コールフロー

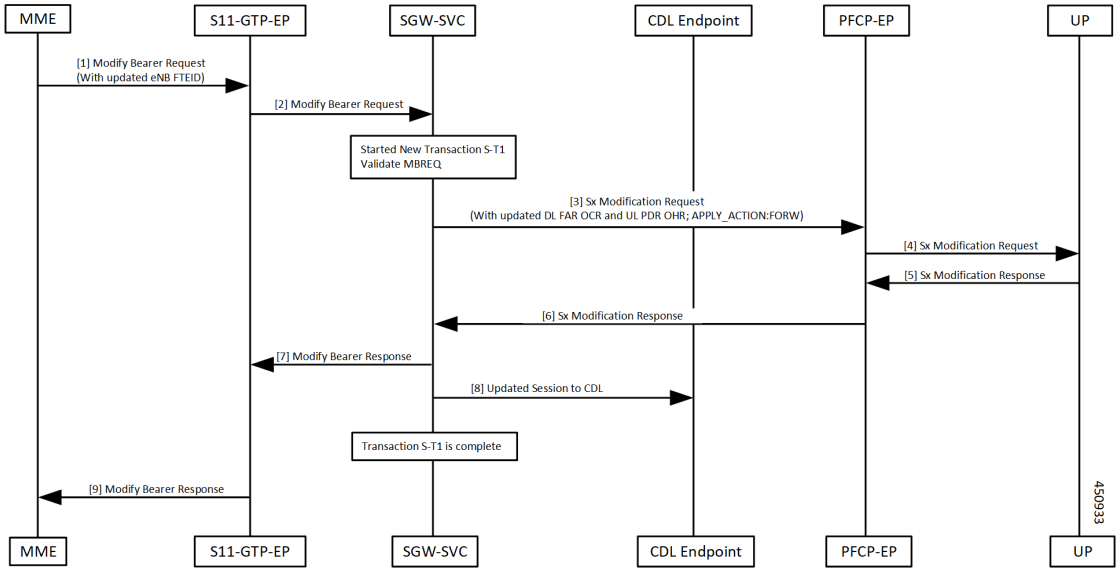


表 *144 : MME* 内ハンドオーバーのアクティブ-アクティブ移行コールフローの説明

ステップ	説明
1	MME が、更新された eNodeB F-TEID とともに、S11-GTP-EP ポッドにベアラー変更要求を送信します。
2	S11-GTP-EP ポッドが SGW-SVC にベアラー変更要求を送信します。 SGW-SVC は次を実行します。 • 新しいトランザクション S-T1 を作成 • ベアラー変更要求を検証
3	SGW-SVC が、ダウンリンク FAR OCR、アップリンク PDR OHR、および Apply Action を FORWARD として、Sx 変更要求を PFCP-EP ポッドに送信します。
4	PFCP-EP ポッドが Sx 変更要求を UP に転送します。
5	UP が Sx 変更応答を PFCP-EP ポッドに送信します。
6	PFCP-EP ポッドが SGW-SVC に Sx 変更応答を送信します。
7	SGW-SVC が S11-GTP-EP ポッドにベアラー変更応答を送信します。
8	SGW-SVC が更新されたセッションを CDL エンドポイントに送信します。 トランザクション S-T1 が完了しました。
9	S11-GTP-EP ポッドが MME にベアラー変更応答を送信します。

MME 間/内ハンドオーバーのアイドル/アイドル移行コールフロー

ここでは、MME 間/内ハンドオーバーのアイドル/アイドル移行コールフローについて説明します。

図 *73 : MME* 間/内ハンドオーバーのアイドル/アイドル移行コールフロー

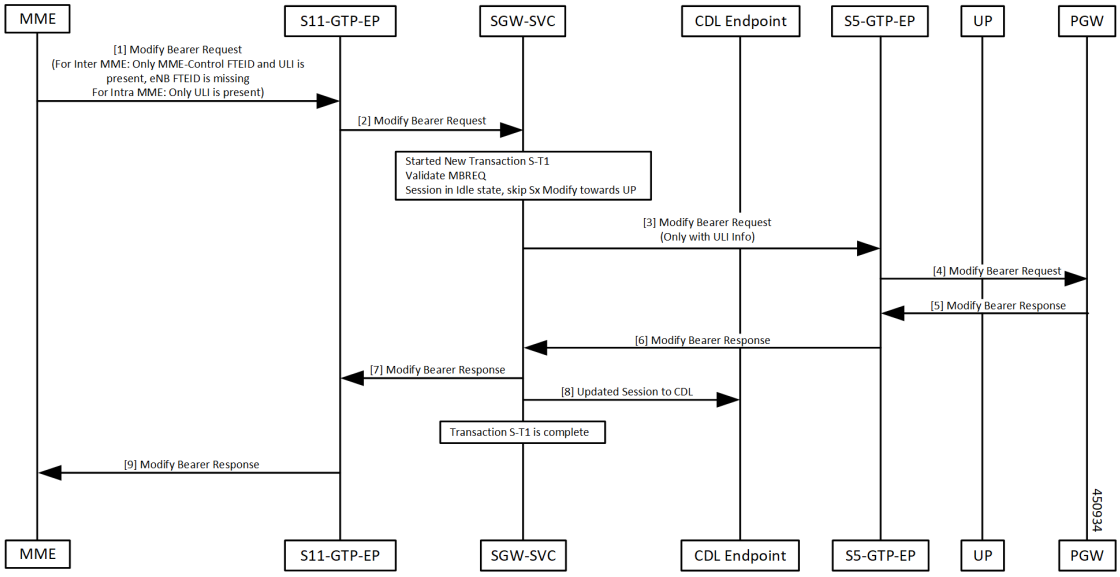


表 *145 : MME* 間/内ハンドオーバーのアイドル/アイドル移行コールフローの説明

ステップ	説明
1、2	MME 間の場合、MME 制御 F-TEID と ULI を使用する S11-GTPC-EP ポッドを介して SGW サービスポッドで Modify Bearer Request を受信します。eNodeB F-TEID はありません。 MME 内の場合、次の ULI のみが存在する SGW サービスポッドで Modify Bearer Request を受信します。 • 新しいトランザクション S-T1 を作成します。 • Modify Bearer Request を検証します。 • セッションがアイドル状態であるため、Sx の変更をスキップします。
3、4	S5 GTPC-EP ポッドで ULI を使用する SGW サービスポッドから Modify Bearer Request を受信します。Modify Bearer Request が PGW に転送されます。
5、6	S5 GTPC-EP ポッドで PGW から Modify Bearer Response を受信します。 SGW サービスポッドで Modify Bearer Response を受信します。

ステップ	説明
7	SGW サービスポッドが Modify Bearer Response を S11 GTPC-EP ポッド入力に転送します。
8、9	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。S11 GTPC-EP ポッドが Modify Bearer Response を MME に転送します。

MME 間/MME 内ハンドオーバーのアクティブ-アイドル移行コールフロー

このセクションでは、MME 間/MME 内ハンドオーバーのアクティブ-アイドル移行コールフローについて説明します。

図 *74 : MME* 間/*MME* 内ハンドオーバーのアクティブ-アイドル移行コールフロー

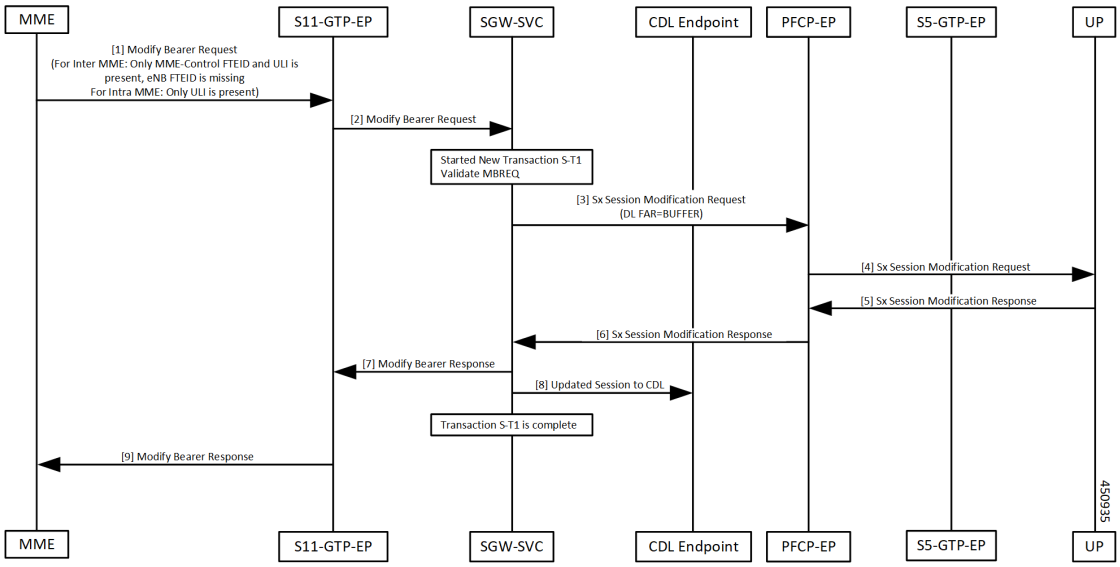


表 *146 : MME* 間/*MME* 内ハンドオーバーのアクティブ-アイドル移行コールフローの説明

ステップ	説明
1、2	MME 間の場合、MME 制御 F-TEID と ULI を使用する S11-GTPC-EP ポッドを介して SGW サービスポッドでベアラー変更要求を受信します。eNodeB F-TEID はありません。 MME 内の場合、次の ULI のみが存在する SGW サービスポッドでベアラー変更要求を受信します。 • 新しいトランザクション S-T1 を作成します。 • ベアラー変更要求を検証します。

ステップ	説明
3、4	PFCP-EP POD で SGW サービスポッドからダウンリンク FAR を BUFFER として、Sx 変更要求を受信しました。Sx 変更要求が UP に転送されます。
5、6	PFCP-EP POD で UP から Sx 変更応答を受信しました。SGW サービスポッドで Sx 変更応答を受信しました。
7	SGW サービスポッドが、ベアラー変更応答を GTPC-EP ポッド入力に転送します。
8、9	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。GTPC-EP POD はベアラー応答の変更を MME に転送します。

MME 間ハンドオーバーおよびマルチ PDN 処理における、ULI 変更を伴うアクティブ-アイドル移行のコールフロー

このセクションでは、MME 間ハンドオーバーおよびマルチ PDN 処理における、ULI 変更を伴うアクティブ-アイドル移行のコールフローについて説明します。

図 *75 : MME* 間ハンドオーバーおよびマルチ *PDN* 処理における、*ULI* 変更を伴うアクティブ-アイドル移行のコールフロー

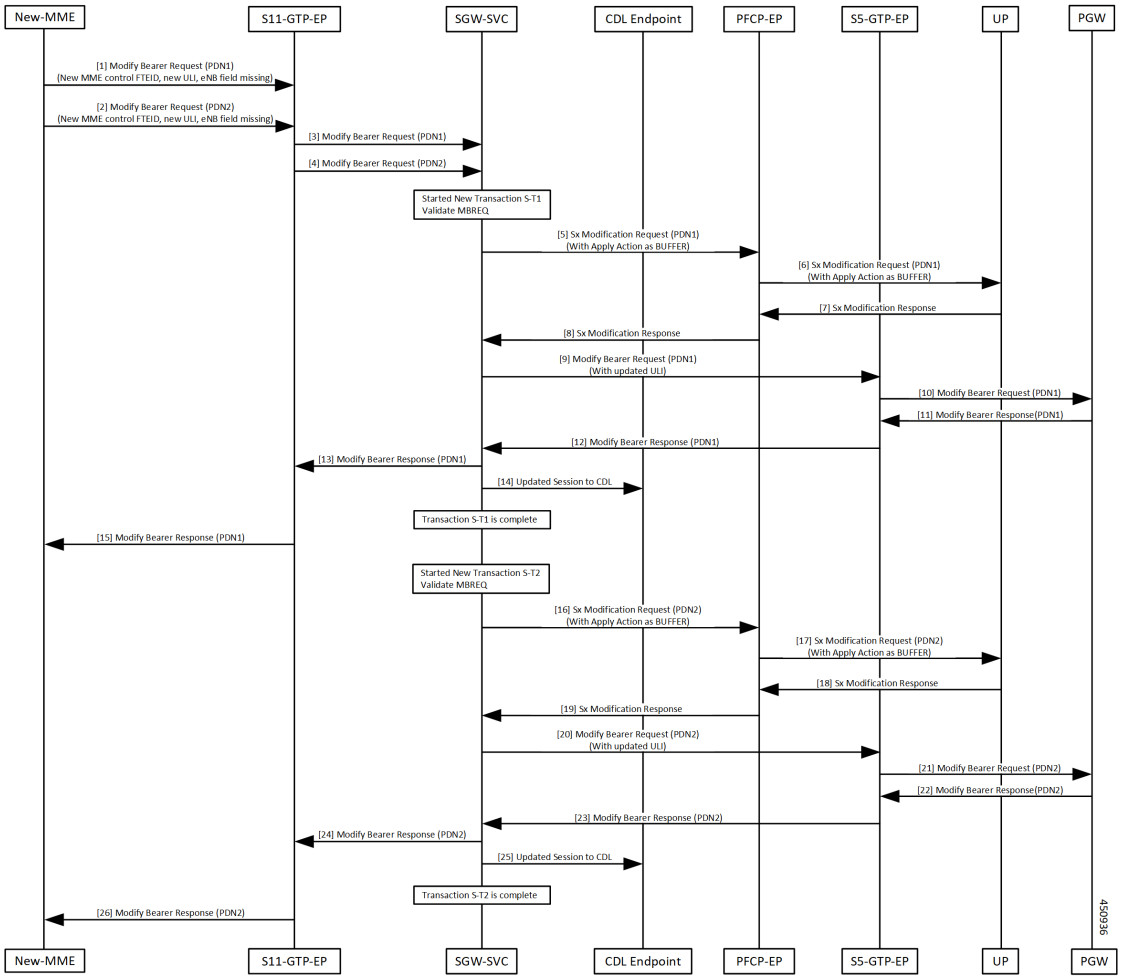


表 147 : MME 間ハンドオーバーおよびマルチ PDN 処理における、ULI 変更を伴うアクティブ-アイドル移行のコールフローの説明 （384 ページ） で提供されている手順を、PDN1 と PDN2 について、それぞれのトランザクション S-T1 と S-T2 で繰り返します。

表 *147 : MME* 間ハンドオーバーおよびマルチ *PDN* 処理における、*ULI* 変更を伴うアクティブ-アイドル移行のコールフローの説明

ステップ	説明
1、2、3、4	新しい MME 制御 F-TEID と新しい ULI とともに、SGW サービスポッドで両方の PDN（PDN1 と PDN2）に対するベアラー変更要求を受信しました。ベアラー変更要求に eNodeB F-TEID がありません。 • 新しいトランザクション S-T1 を作成します。 • ベアラー変更要求を検証します。

ステップ	説明
5、6	Apply Action を BUFFER として、SGW サービスポッドから PFCP-EP ポッドへの Sx 変更要求を受信しました。 Sx 変更要求が UP に転送されます。
7、8	UP から PFCP-EP ポッドへの Sx 変更応答を受信しました。 SGW サービスポッドで Sx 変更応答を受信しました。
9、10	更新された ULI とともに、SGW サービスポッド から S5 GTPC-EP ポッドへのベアラー変更要求を受信しました。 ベアラー変更要求は PGW で受信されます。
11、12	PGW から S5 GTPC-EP ポッドへのベアラー変更応答を受信しました。 S5 GTPC-EP ポッドから SGW サービスポッドへのベアラー変更応答を受信しました。
13	SGW サービスポッドが、ベアラー変更応答を GTPC-EP ポッド入力に転送します。
14、15	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。GTPC-EP ポッドはベアラー変更応答を MME に転送します。

削除コールフローとしてマークされたベアラーコンテキストを使用した MME 間ハンドオーバー

このセクションでは、削除コールフローとしてマークされたベアラーコンテキストを使用した MME 間ハンドオーバーのコールフローについて説明します。

図 *76 :* 削除コールフローとしてマークされたベアラーコンテキストを使用した *MME* 間ハンドオーバー

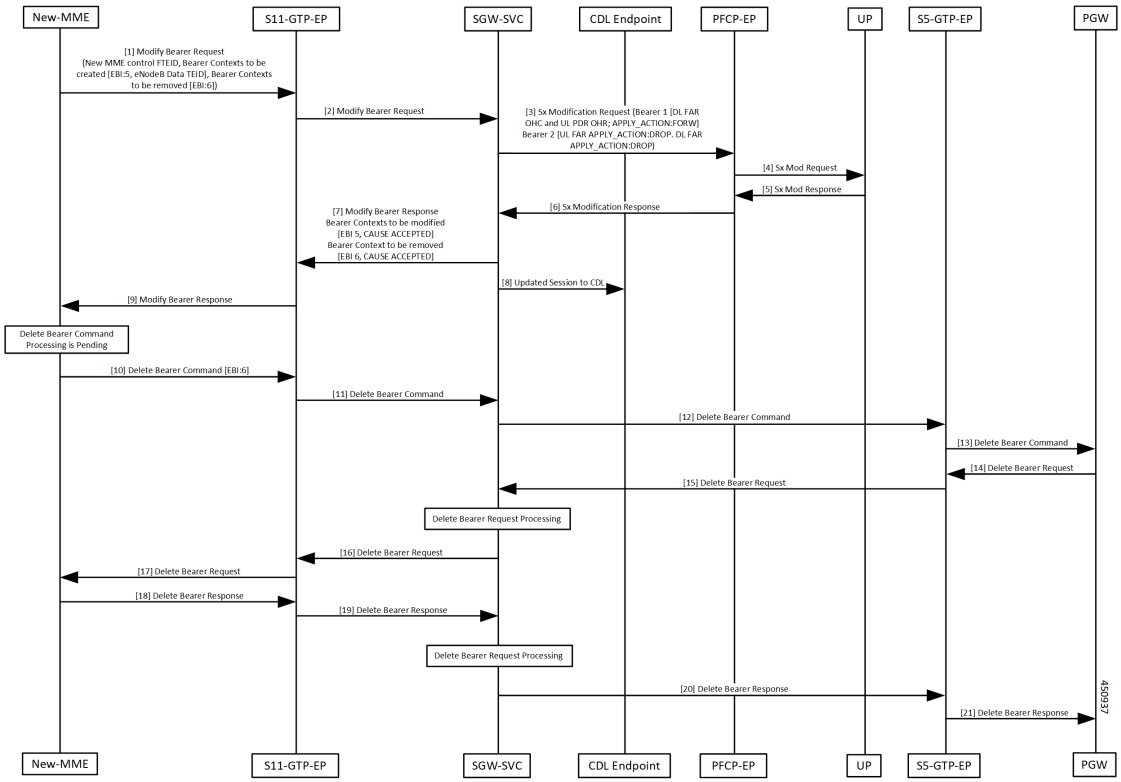


表 *148 :* 削除コールフローとしてマークされたベアラーコンテキストを使用した *MME* 間ハンドオーバーの説明

ステップ	説明
1、2	EBI:5 で作成される新しい MME 制御 F-TEID、bearer-context-1、eNodeB データ TEID、EBI:6 で削除される Bearer-context-2 を持つ SGW-Service POD でベアラー変更要求を受信しました。 • 新しいトランザクション S-T1 を作成します。 • ベアラー変更要求を検証します。
3、4	SGW サービスポッドからPFCP-EP PODへの次の Sx 変更要求を受信しました。 • Bearer-context-1：ダウンリンク FAR OHC、アップリンク PDR OHR、APPLY ACTION を FORWARD として • Bearer-context-2：アップリンク FAR APPLY ACTION を DROP として、ダウンリンク FAR APPLY ACTION を DROP として Sx 変更要求が UP に転送されます。

ステップ	説明
5、6	UP > PFCP-EP POD から Sx 変更応答を受信しました。 SGW サービスポッドで Sx 変更応答を受信しました。
7	SGW サービスポッドは、変更対象のベアラーコンテキスト（EBI:5、原因として ACCEPTED）と削除対象のベアラーコンテキスト（EBI:6、原因として ACCEPTED）とともにベアラー変更応答を GTPC-EP POD 入力に転送します。
8、9	セッションが CDL で更新されました。GTPC-EP POD はベアラー変更応答を MME に転送します。
10、11	SGW サービスポッドで EBI:6 のベアラー削除コマンドを受信しました。
12、13	SGW サービスポッドが、ベアラー削除コマンドを S5 GTPC-EP POD に転送します。 PGW でベアラー削除コマンドを受信しました。
14、15	PGW > S5 GTPC-EP POD からベアラー削除要求を受信しました。 SGW-Service POD でベアラー削除要求を受信しました。
16、17	SGW サービスポッドがベアラー削除要求を処理し、GTPC-EP POD の入力に転送します。
18、19	SGW-Service POD でベアラー削除応答を受信し、処理しました。
20、21	S5 GTPC-EP POD でベアラー削除応答を受信しました。 PGW でベアラー削除応答を受信しました。

MME 内および MME 間ハンドオーバーの手順の OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

MME 内および MME 間ハンドオーバー手順の機能では、次の統計情報がサポートされています。

MME 内ハンドオーバー

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",
instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
```

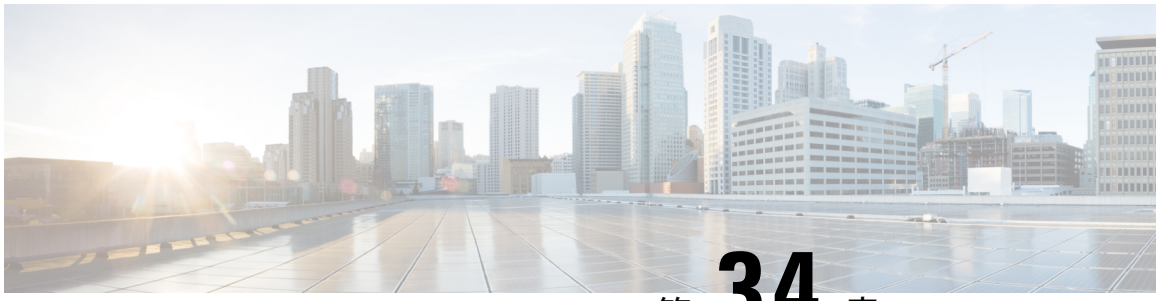
```
 sgw_procedure_type="intra_mme_handover",status="attempted",sub_fail_reason=""}
 2
```

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="intra_mme_handover",status="success",sub_fail_reason=""}
 2
```

MME 間ハンドオーバー

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="inter_mme_handover",status="attempted",sub_fail_reason=""}
 2
```

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="",

instance_id="0",interface="interface_sgw_ingress",reject_cause="",service_name="sgw-service",
 sgw_procedure_type="inter_mme_handover",status="success",sub_fail_reason=""}
 2 Perform S1 Based SGW handover (with OI=0)
```



第 34 章

SGW サービスでの MCC/MNC 構成

- 機能の概要と変更履歴 （389 ページ）
- 機能説明 （390 ページ）
- 機能の仕組み （390 ページ）
- SGW サービスでの MCC または MNC の設定 （391 ページ）
- OAM サポート （392 ページ）

機能の概要と変更履歴

要約データ

表 *149* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	[Disabled]：有効にするには設定が必要
関連資料	該当なし

マニュアルの変更履歴

表 *150* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

この機能は、SGW プロファイルの PLMN リストとして MCC および MNC 設定のリストをサポートします。

SGW プロファイルの PLMN リスト設定に従って、PLMN タイプは次のいずれかとして識別されます。

- ホーマー
- ローマー
- ビジター

（注） この機能が有効になっていない場合、PLMN タイプのサブスクライバはデフォルトでビジターとしてマークされます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

PLMN タイプの検出コールフロー

このセクションでは、PLMN タイプの検出コールフローについて説明します。

図 *77 : PLMN* タイプの検出コールフロー

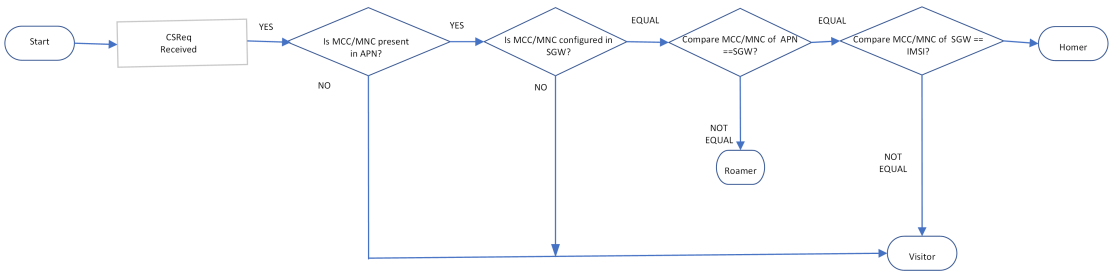


表 *151 : PLMN* タイプの検出コールフローの説明

ステップ	説明
1	PLMN タイプの検出プロセスが開始されます。
2	セッション作成要求を受信しました。
3	APN に MCC または MNC が存在することが確認されます。 • 「YES」の場合、次のステップが実行されます。 • 「NO」の場合、PLMN タイプのサブスクライバは「Visitor」と契約されます。
4	SGW 内の MCC または MNC の有無が確認された場合： • 「YES」の場合、次のステップが実行されます。 • 「NO」の場合、PLMN タイプのサブスクライバは「Visitor」と契約されます。
5	APN と SGW の MCC または MNC の値が比較されます。 • 「EQUAL」の場合、次のステップに進みます。 • 「NOT EQUAL」の場合、PLMN タイプのサブスクライバはローミングと結論付けられます。
6	SGW と IMSI の MCC または MNC の値が比較されます。 • 「EQUAL」の場合、PLMN タイプのサブスクライバは Homer として確定されます。 • 「NOT EQUAL」の場合、PLMN タイプのサブスクライバはビジターとして確定されます。

SGW サービスでの MCC または MNC の設定

ここでは、SGW サービスでの MCC または MNC の設定方法について説明します。

次のコマンドを使用して、SGW サービスで MCC または MNC を設定します。

```
config
   profile sgw sgw_name
      plmn-list
         mcc mcc_value
         mnc mnc_value
         end
```

注：

- **plmn-list**：MCC および MNC 値のリスト。
- **mcc** *mcc_value*：MCC 値を指定します。3 桁の数字で指定する必要があります。例：123
- **mnc** *mnc_value*：MNC 値を指定します。2 桁または 3 桁の数字で指定する必要があります。例：23 または 456

設定例

次に、設定例を示します。

```
config
   profile sgw sgw_1
      plmn-list
         mcc 123
         mnc 456
         end
```

OAM サポート

このセクションでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

SGW サービス機能の MCC および MNC 設定では、次の統計がサポートされています。

アクティブ PDN カウンタ

```
sgw_pdn_counters{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="homer",pdn_type="ipv6",rat_type="EUTRAN",
service_name="sgw-service"} 12
```

```
sgw_pdn_counters{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="roamer",pdn_type="ipv4v6",rat_type="EUTRAN",
service_name="sgw-service"} 3
```

```
sgw_pdn_counters{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="visitor",pdn_type="ipv4",rat_type="EUTRAN",
service_name="sgw-service"} 2
```

設定またはリリースされた PDN の統計

```
sgw_pdn_stats{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="homer",pdn_type="ipv6",rat_type="EUTRAN",
service_name="sgw-service",status="release"} 1
```

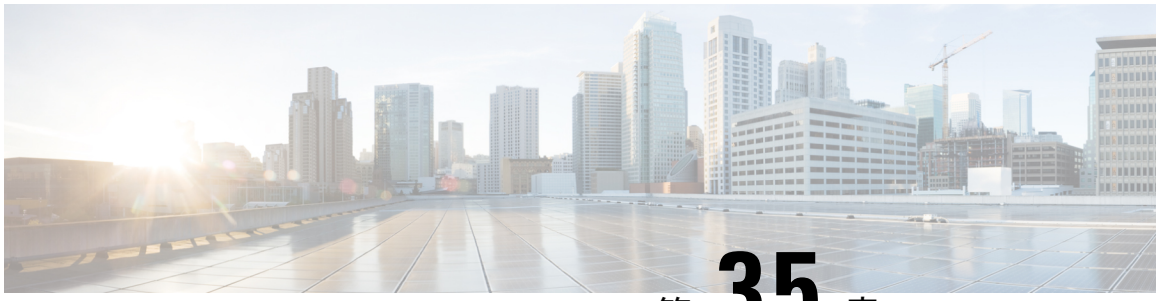
```
sgw_pdn_stats{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="homer",pdn_type="ipv6",rat_type="EUTRAN",
service_name="sgw-service",status="setup"} 13
```

```
sgw_pdn_stats{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="roamer",pdn_type="ipv4v6"rat_type="EUTRAN",
service_name="sgw-service",status="release"} 1
```

```
sgw_pdn_stats{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="roamer",pdn_type="ipv4v6",rat_type="EUTRAN",
service_name="sgw-service",status="setup"} 4
```

```
sgw_pdn_stats{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="visitor",pdn_type="ipv4",rat_type="EUTRAN",
service_name="sgw-service",status="release"} 1
```

```
sgw_pdn_stats{app_name="smf",cluster="cn",data_center="cn",
instance_id="0",pdn_plmn_type="visitor"pdn_type="ipv4",rat_type="EUTRAN",
service_name="sgw-service",status="setup"} 3
```



第 35 章

メッセージインタラクションのサポート

- 機能の概要と変更履歴 （395 ページ）
- 機能説明 （396 ページ）
- 機能の仕組み （397 ページ）

機能の概要と変更履歴

要約データ

表 *152 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	[Enabled]：常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *153* : マニュアルの変更履歴

改訂の詳細	リリース
次の手順が追加されました。 • コリジョンリゾルバ • 複数の CBR • 二重削除の最適化 • 優先順位が低い場合の処理の中止と一時停止の処理	2021.02.0
最初の導入。	2021.01.0

機能説明

メッセージインタラクション機能は、さまざまなピア（UPF、MME、およびPGW）からのメッセージを受信および処理する機能を提供し、優先順位の解決を実行します。

次に、メッセージインタラクションのシナリオと優先順位の例を示します。

- 同じ PDN1 に対して受信した Modify Bearer Request（MBR）と Update Bearer Request（UBR）は、UPF からの Sx Modify Request を待機します。UBR は MBR の完了後に処理されます。このシナリオでは、MBR が処理されるまで UBR プロセスは一時停止されます。
- 同じ PDN1 の Release Access Bearer（RAB）の処理中に、PDN1 の UBR1 を受信した場合、UBR は RAB の完了後に処理されます。このシナリオでは、RAB 手順の進行中に UBR1 プロセスが一時停止されます。
- 既存の PDN 手順は、切断手順（Delete Session Request（DSR）、Delete Bearer Request（DBR）、または Clear Sub）が同じ PDN に対して送信されたときに停止します。たとえば、同じ PDN1 に対して CBR1 手順と UBR1 手順が進行中の場合、cnSGW-C は PDN の DSR を受信します。DSR 処理が開始され、CBR1 および UBR1 処理が停止します。詳細については、既存の PDN 手順のグレースフル停止のコールフロー （400 ページ）コールフローを参照してください。
- 既存の UE 手順（RAB または DDN）は、PDN の切断手順（DSR、DBR、または Clear Sub）を受信すると停止します。たとえば、cnSGW-C は、RAB または DDN の処理中に PDN の DBR を受信した場合、DBR 手順が開始し、RAB または DDN 手順が停止します。
- 同じ PDN の切断手順（DSR、DBR、または Clear Sub）が進行中の場合、PDN の着信手順は停止します。たとえば、UBR が同じ PDN の DSR を送信する際に PDN を受信する場合、UBR 手順が停止し、DSR 手順が続行されます。

- 同じ PDN の切断手順（DSR、DBR、または Clear Sub）を処理している場合、新しい着信 UE 手順が停止します。たとえば、同じ PDN1 のマルチ PDN コール DSR を処理するときに、PDN1 のRAB を受信した場合。RAB 手順は、PDN1 の DSR が完了後に停止し、再スケジュールされます。
- 初期接続手順が進行中の場合、CBR および UBR メッセージ処理は停止します。
- 二重削除処理の最適化。たとえば、cnSGW-C は、UPF に対する DBR Sx 変更手順が保留中の場合、PGW の MME および DBR から DSR を受信し、DBR シグナリングは S11 インターフェイスに対して開始されません。
- 優先順位が高い手順を同じベアラーで受信すると、優先順位が低い手順の処理が停止します。たとえば、cnSGW-C が、同じベアラーで UBR を処理しながら、専用のベアラーで PDN の DBR を受信する場合、DBR 処理手順が開始し、同じベアラーにおける既存の UBR 手順の処理が停止します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

CBR Multi-PDN コールフロー

このセクションでは、CBR Multi-PDN コールフローについて説明します。

図 *78 : CBR Multi-PDN* コールフロー

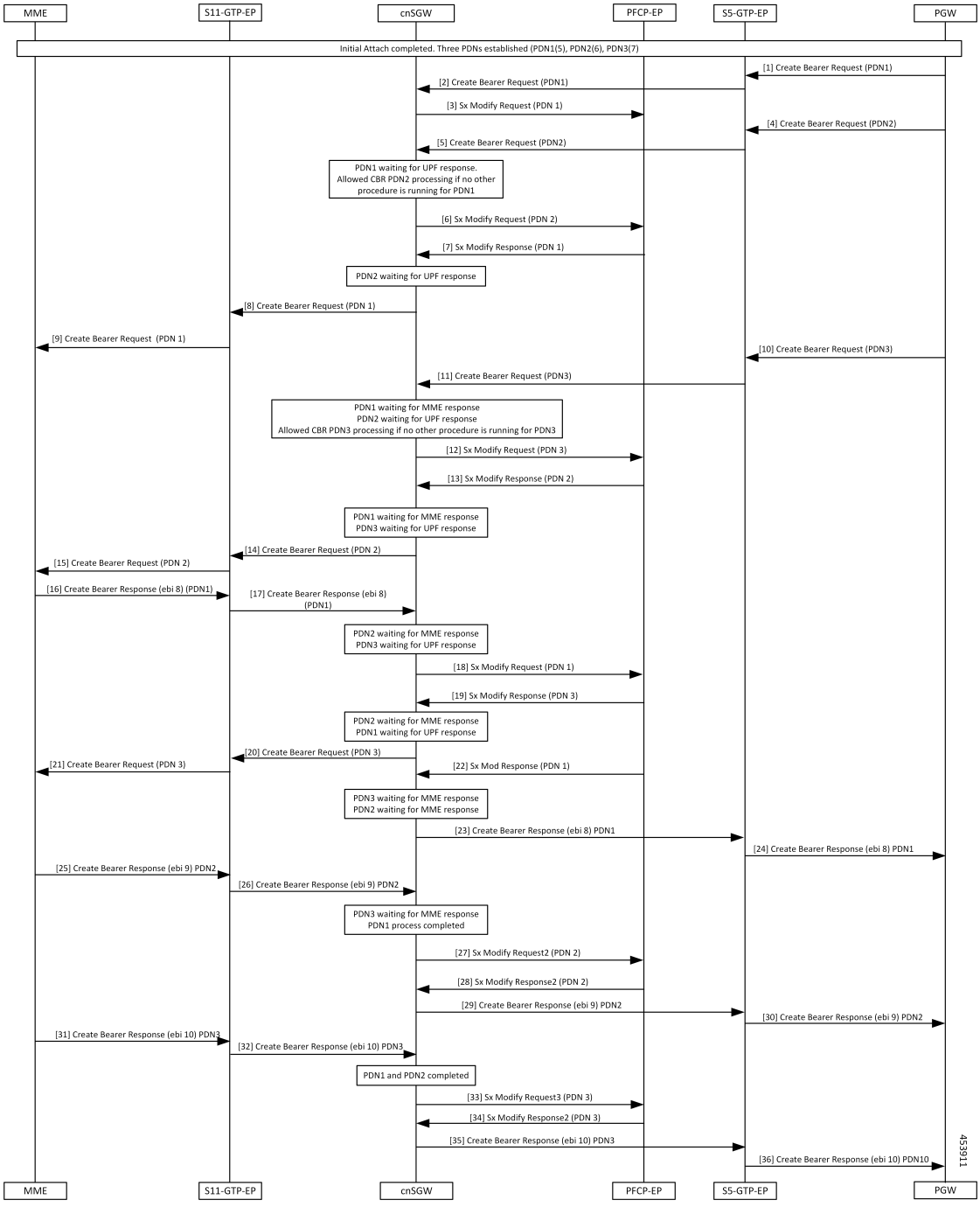


表 *154 : CBR Multi-PDN* コールフローの説明

ステップ	説明
1	3 つの確立済み PDN により、初期接続プロセスが完了します。 PGW が S5-GTP-EP に PDN1 のベアラー作成要求を送信します。
2	S5-GTP-EP が cnSGW に PDN1 のベアラー作成要求を送信します。
3	cnSGW が PFCP-EP に PDN1 の Sx 変更要求を送信します。
4	PGW が S5-GTP-EP に PDN2 のベアラー作成要求を送信します。
5	S5-GTP-EP が cnSGW にベアラー作成要求 PDN2 を送信します。
6	cnSGW が PFCP-EP に PDN2 の Sx 変更要求を送信します。
7	PFCP-EP が cnSGW に PDN1 の Sx 変更応答を送信します。
8	cnSGW が S11-GTP-EP に PDN1 のベアラー作成要求を送信します。
9	S11-GTP-EP が MME に PDN1 のベアラー作成要求を送信します。
10	PGW が S5-GTP-EP にベアラー作成要求（PDN3）を送信します。
11	S5-GTP-EP が cnSGW にベアラー作成要求（PDN3）を送信します。
12	cnSGW が PFCP-EP に Sx 変更要求（PDN3）を送信します。
13	PFCP-EP が cnSGW に PDN2 の Sx 変更応答を送信します。
14	cnSGW が S11-GTP-EP に PDN2 のベアラー作成要求を送信します。
15	S11-GTP-EP が MME に PDN1 のベアラー作成要求を送信します。
16	MME が S11-GTP-EP に PDN1 のベアラー作成応答を送信します。
17	S11-GTP-EP が cnSGW に PDN1 のベアラー作成応答を送信します。
18	cnSGW が PFCP-EP に PDN1 の Sx 変更要求を送信します。
19	PFCP-EP が cnSGW に PDN3 の Sx 変更応答を送信します。
20	cnSGW が S11-GTPC-EP に PDN3 のベアラー作成要求を送信します。
21	S11-GTPC-EP が MME に PDN3 のベアラー作成要求を送信します。
22	PFCP-EP が cnSGW に PDN1 の Sx 変更応答 2 を送信します。
23	cnSGW が S5-GTP-EP に PDN1 のベアラー作成応答を送信します。

ステップ	説明
24	S5-GTP-EP が PGW に PDN1 のベアラー作成応答を送信します。
25	MME が S11-GTP-EP に PDN2 のベアラー作成応答を送信します。
26	S11-GTPC-EP が cnSGW に PDN1 のベアラー作成応答を送信します。
27	cnSGW が PFCP-EP に PDN2 の Sx 変更要求を送信します。
28	PFCP-EP が cnSGW に PDN2 の Sx 変更応答を送信します。
29	cnSGW が S5-GTP-EP に PDN2 のベアラー作成応答を送信します。
30	S5-GTP-EP が PGW に PDN2 のベアラー作成応答を送信します。
31	MME が S11-GTP-EP に PDN3 のベアラー作成応答を送信します。
32	S11-GTPC-EP が cnSGW に PDN3 のベアラー作成応答を送信します。
33	cnSGW が PFCP-EP に PDN2 の Sx 変更要求を送信します。
34	PFCP-EP が cnSGW に PDN3 の Sx 変更応答を送信します。
35	cnSGW が S5-GTP-EP に PDN3 のベアラー作成応答を送信します。
36	S5-GTP-EP が PGW に PDN3 のベアラー作成応答を送信します。

既存の PDN 手順のグレースフル停止のコールフロー

ここでは、既存の PDN 手順のグレースフル停止のコールフローについて説明します。

図 *79 :* 既存の *PDN* 手順のグレースフル停止のコールフロー

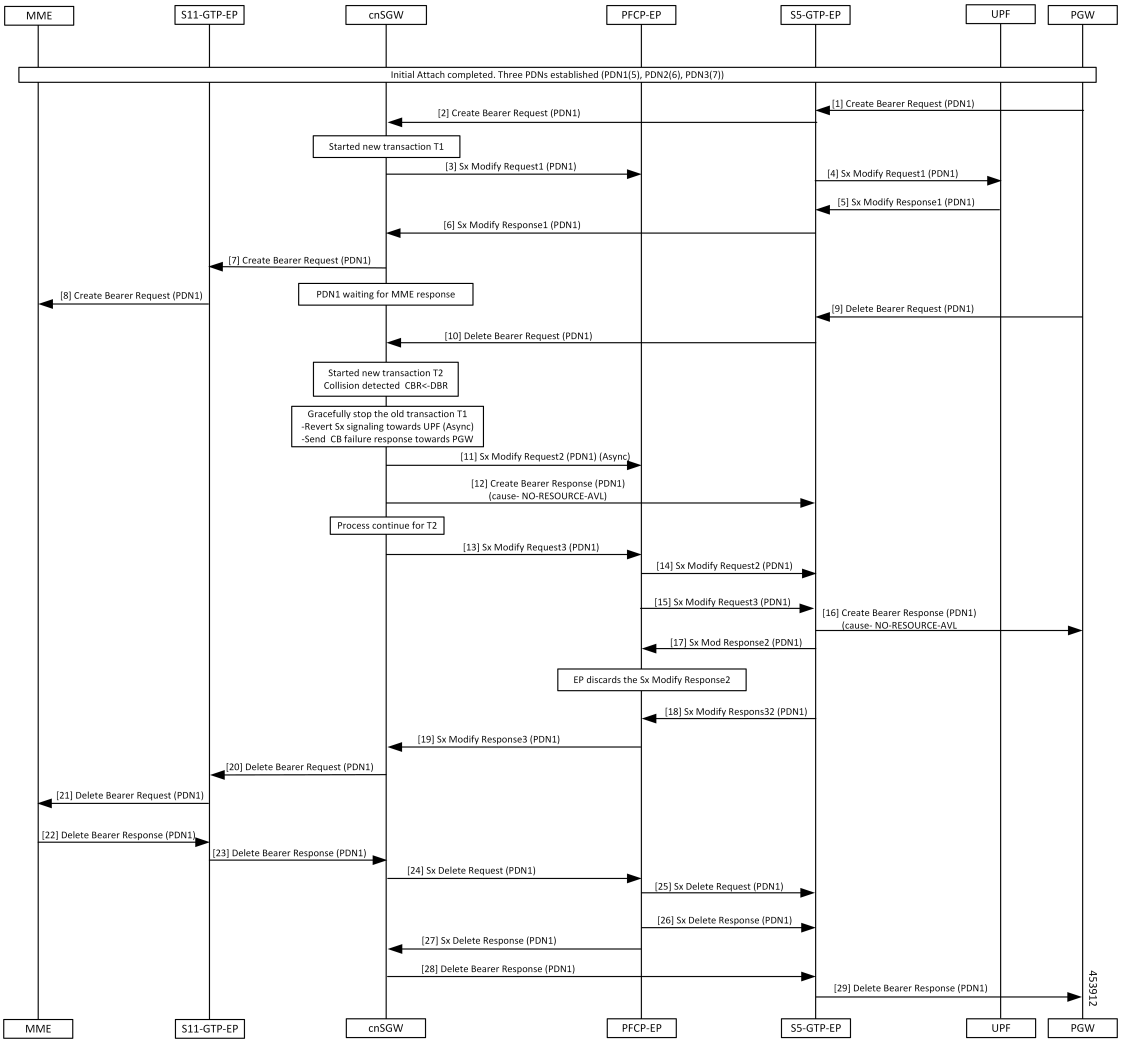


表 *155 :* 既存の *PDN* 手順のグレースフル停止のコールフローの説明

ステップ	説明
1	すでに確立されている 3 つの PDN により、初期接続プロセスが完了します。 PGW が S5-GTPC-EP にベアラー作成要求を送信します。
2	S5-GTCP-EP が cnSGW に PDN1 のベアラー作成要求を転送します。
3	cnSGW が PFCP-EP に PDN1 の Sx 変更要求を転送します。 cnSGW は PDN1 の Sx 変更応答を待機します。
4	PFCP-EP が UPF に PDN1 の Sx 変更要求を送信します。

ステップ	説明
5	UPF が PFCP-EP に PDN1 の Sx 変更応答を送信します。
6	PFCP-EP が cnSGW に PDN1 の Sx 変更応答を送信します。
7	cnSGW が S11-GTPC-EP に PDN1 のベアラー作成要求を送信します。
8	S11-GTPC-EP が MME に PDN1 のベアラー作成要求を転送します。
9	PGW が S5-GTPC-EP に PDN1 のベアラー削除要求を送信します。
10	S5-GTPC-EP が cnSGW に PDN1 のベアラー削除要求を転送します。 cnSGW は MME 応答を待ち、PDN1 のベアラー削除要求を受信します。 PDN1 のコリジョンが検出された場合は、PDN1 の古いトランザクション T1 を停止します。
11	cnSGW が PFCP-EP に Sx 変更要求（非同期）を送信します。
12	cnSGWcnSGW が、No-Resource-Available を原因として PDN1 のベアラー作成応答を S5-GTPC-EP に送信します。
13	cnSGW が PFCP-EP に DBR の Sx 変更要求を送信します。
14	PFCP-EP が UPF に CBR の Sx 変更要求を転送します。
15	PFCP-EP が UPF に DBR の Sx 変更要求を転送します。
16	S5-GTPC-EP が、No を原因としてベアラー作成応答を送信します。
17	UPF が PFCP-EP に CBR の Sx 変更応答を送信します。 PFCP-EP はこの応答を破棄します。
18	UPF が PFCP-EP に DBR の Sx 変更応答を送信します。
19	PFCP-EP が cnSGW に DBR の Sx 変更応答を転送します。
20	cnSGW が S11-GTPC-EP に PDN1 のベアラー削除要求を送信します。
21	S11-GTPc-EP が MME に PDN1 のベアラー削除要求を転送します。
22	MME が S11-GTPC-EP に PDN1 のベアラー削除応答を送信します。
23	S11-GTCP-EP が cnSGW に PDN1 のベアラー削除応答を転送します。
24	cnSGW が PFCP-EP に PDN1 の Sx 削除要求を送信します。
25	PFCP-EP が UPF に PDN1 の Sx 削除要求を転送します。

ステップ	説明
26	UPF が PFCP-EP に PDN1 の Sx 削除応答を送信します。
27	PFCP-EP が cnSGW に PDN1 の Sx 削除応答を転送します。
28	cnSGW が S5-GTPC-EP に PDN1 のベアラー削除応答を送信します。
29	S5-GTPC-EP が PGW に PDN1 のベアラー削除応答を転送します。

マルチ PDN 処理（PGW インタラクションあり）を使用した MME 間ハンドオーバーコールフロー

ここでは、マルチ PDN 処理（PGW インタラクションあり）を使用した MME 間ハンドオーバーコールフローについて説明します。

図 *80 :* マルチ *PDN* 処理（*PGW* インタラクションあり）を使用した *MME* 間ハンドオーバーコールフロー

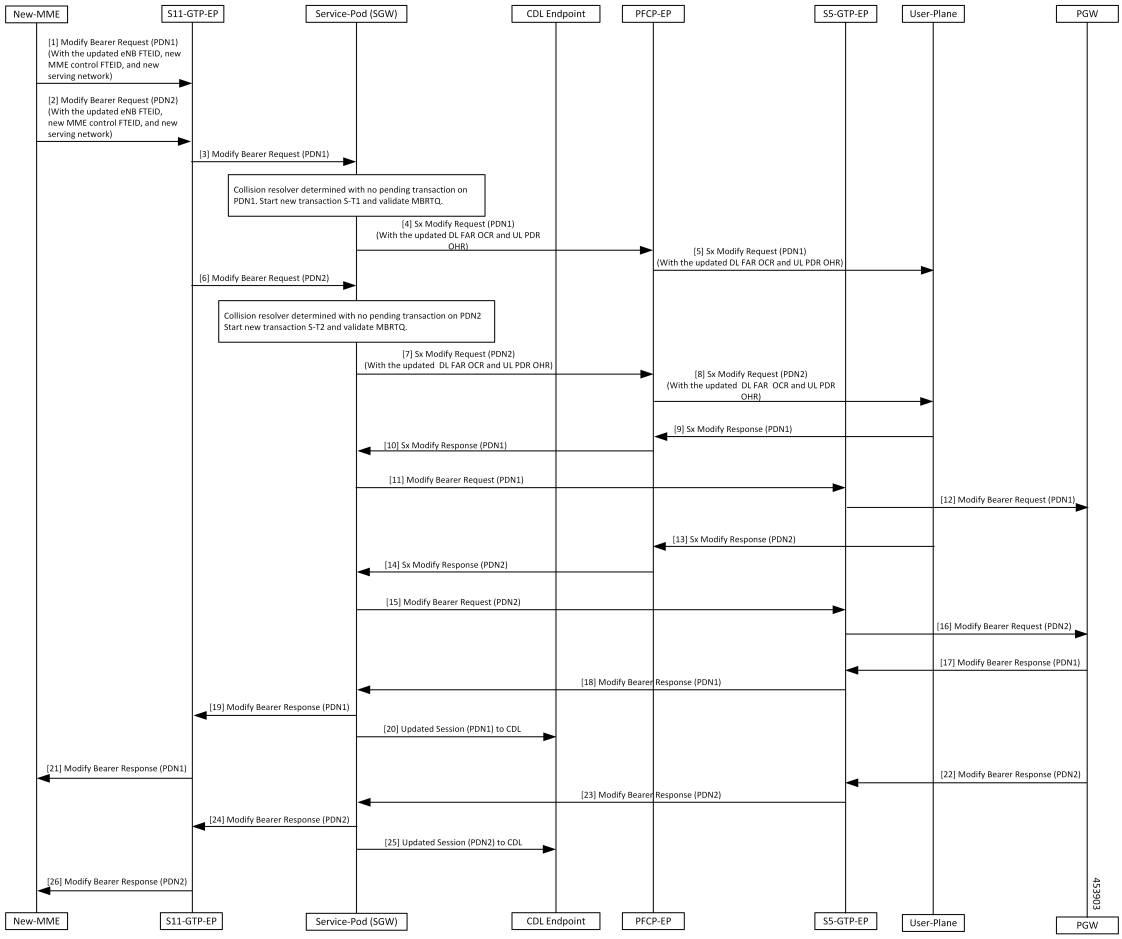


表 *156 :* マルチ *PDN* 処理（*PGW* インタラクションあり）を使用した *MME* 間ハンドオーバーコールフローの説明

ステップ	説明
1	New-MME が更新された eNodeB FTEID、新しい MME 制御 FTEID、新しいサービングネットワークを使用して、（PDN1）の Modify Bearer Request を S11-GTP-EP に送信します。
2	New-MME が更新された eNodeB FTEID、新しい MME 制御 FTEID、新しいサービングネットワークを使用して、（PDN2）の Modify Bearer Request を S11-GTP-EP に送信します。
3	S11-GTP-EP が Modify Bearer Request（PDN1）を cnSGW-C に送信します。
4	cnSGW-C が更新された DL FAR OCR と UL PDR OHR を含む Sx Modify Request（PDN1）を PFCP-EP に送信します。
5	PFCP-EP が（PDN1）の Sx Modify Request をユーザープレーンに転送します。
6	S11-GTP-EP が Modify Bearer Request（PDN2）を送信します。
7	サービスポッド（SGW）が更新された DL FAR OCR と UL PDR OHR を含む Sx Modify Request（PDN2）を PFCP-EP に送信します。
8	PFCP-EP が Sx Modify Request（PDN2）をユーザープレーンに転送します。
9	ユーザープレーンが Sx Modify Request（PDN1）を PFCP-EP に送信します。
10	PFPC-EP が Sx Modify Request（PDN1）をサービスポッド（SGW）に転送します。
11	サービスポッド（SGW）が更新されたサービングネットワーク情報を含む Modify Bearer Request（PDN1）を S5-GTP-EP に送信します。
12	S5-GTP-EP が Modify Bearer Request（PDN1）を PGW に転送します。
13	ユーザープレーンが Sx Modify Request（PDN2）を PFCP-EP に送信します。
14	PFCP-EP が Sx Modify Request（PDN2）をサービスポッド（SGW）に転送します。
15	サービスポッド（SGW）が更新されたサービングネットワーク情報を含む Modify Bearer Request（PDN2）を S5-GTP-EP に送信します。
16	S5-GTP-EP が Modify Bearer Request（PDN2）を PGW に送信します。
17	PGW が Modify Bearer Response（PDN1）を S5-GTP-EP に送信します。
18	S5-GPT-EP が Modify Bearer Response（PDN1）をサービスポッド（SGW）に転送します。

ステップ	説明
19	サービスポッド（SGW）が Modify Bearer Response（PDN1）を S11-GTP-EP に転送します。
20	サービスポッド（SGW）が CDL エンドポイントに送信された PDN1 セッションを更新します。
21	S11-GTP-EP が Modify Bearer Response（PDN1）を New-MME に送信します。
22	PGW が（PDN2）の Modify Bearer Response を S5-GTP-EP に送信します。
23	S5-GPT-EP が Modify Bearer Response（PDN2）をサービスポッド（SGW）に転送します。
24	サービスポッド（SGW）が Modify Bearer Response（PDN2）を S11-GTP-EP に転送します。
25	サービスポッド（SGW）が MME 間を完了（PDN2）とマークし、CDL エンドポイントに送信された PDN2 セッションを更新します。
26	S11-GTP-EP が Modify Bearer Response（PDN2）を New-MME に送信します。

マルチ PDN コール X2 ハンドオーバー SGW の cnSGW-C への再配置コールフロー

ここでは、マルチ PDN コール X2 ハンドオーバー SGW の cnSGW-C への再配置コールフローについて説明します。

図 *81 :* マルチ *PDN* コール *X2* ハンドオーバー *SGW* の *cnSGW-C* への再配置コールフロー

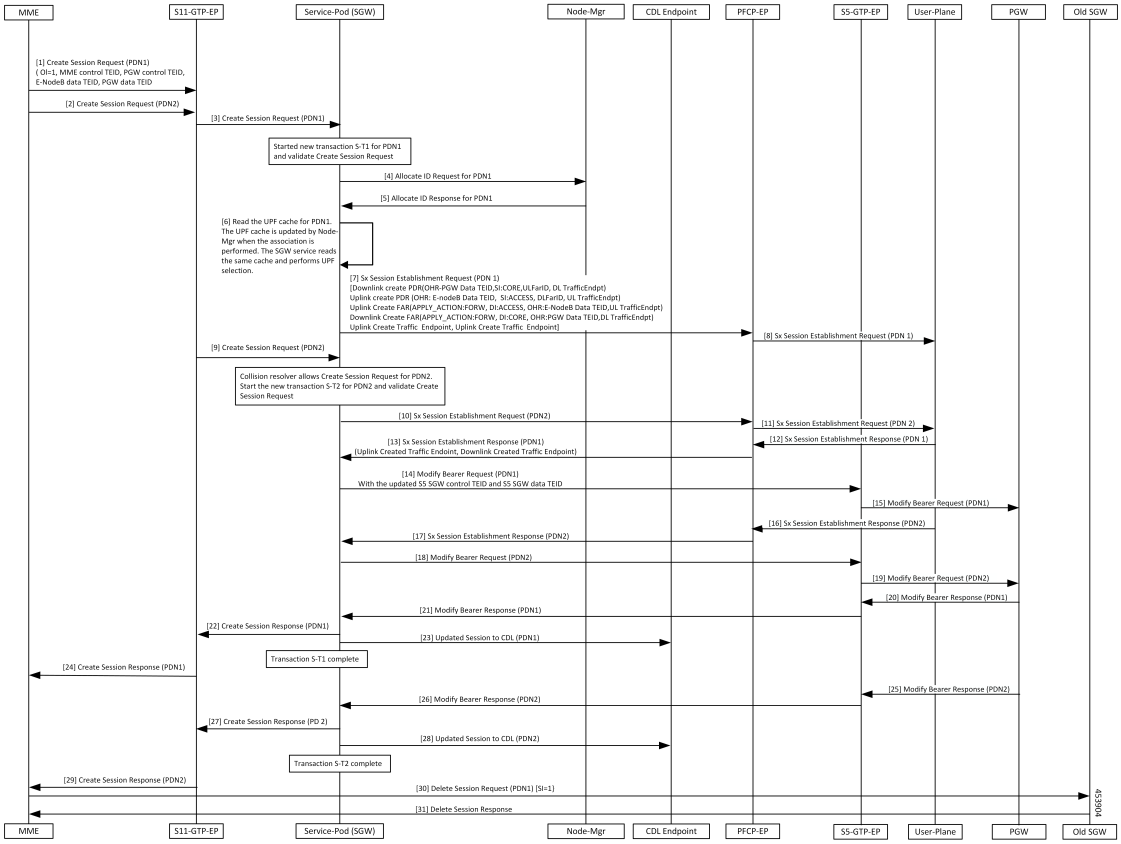


表 *157 :* マルチ *PDN* コール *X2* ハンドオーバー *SGW* の *cnSGW-C* への再配置コールフローの説明

ステップ	説明
1	MME が（OI = 1、MME 制御 TEID、PGW 制御 TEID、eNodeB データ TEID、PGW データ TEID）を使用して（PDN1）の Create Session Request を S11-GTP-EP に送信します。
2	MME が Create Session Request（PDN2）を S11-GTP-EP に送信します。
3	S11-GTP-EP が Create Session Request（PDN1）をサービスポッド（SGW）に転送します。
4	サービスポッド（SGW）が Node-Mgr に ID の割り当て（PDN1）を要求します。
5	Node-Mgr が ID 割り当て応答（PDN1）で応答します。
6	サービスポッド（SGW）が UPF 選択を実行します。
7	サービスポッド（SGW）が Sx Session Establishment Request（PDN1）を PFCP-EP に送信します。

ステップ	説明
8	PFCP-EP が Sx Session Establishment Request（PDN1）をユーザープレーンに転送します。
9	S11-GTP-EP が Create Session Request（PDN2）をサービスポッド（SGW）に送信します。
10	サービスポッド（SGW）が Sx Session Establishment Request（PDN2）を PFCP-EP に送信します。
11	PFCP-EP が Sx Session Establishment Request（PDN2）をユーザープレーンに転送します。
12	ユーザープレーンが Sx Session Establishment Response（PDN1）を PFCP-EP に送信します。
13	PFCP-EP が Sx Session Establishment Response（PDN1）をサービスポッド（SGW）に送信します。
14	サービスポッド（SGW）が、更新済みの S5 SGW 制御 TEID と S5 SGW データ TEID を使用して、Modify Bearer Request（PDN1）を S5-GTP-EP に送信します。
15	S5-GTP-EP が Modify Bearer Request（PDN1）を PGW に転送します。
16	ユーザープレーンが Sx Session Establishment Response（PDN2）を PFCP-EP に送信します。
17	PFCP-EP が Sx Session Establishment Response（PDN2）をサービスポッド（SGW）に転送します。
18	サービスポッド（SGW）が、更新済みの S5 SGW 制御 TEID と S5 SGW データ TEID を使用して、Modify Bearer Request（PDN2）を S5-GTP-EP に送信します。
19	S5-GTP-EP が Modify Bearer Request（PDN2）を PGW に転送します。
20	PGW が Modify Bearer Response（PDN1）を S5-GTP-EP に送信します。
21	S5-GTP-EP が Modify Bearer Response（PDN1）をサービスポッド（SGW）に転送します。
22	サービスポッド（SGW）が Create Session Response（PDN1）を S11-GTP-EP に送信します。
23	サービスポッド（SGW）が CDL エンドポイントに対する PDN1 セッションを更新します。
24	S11-GTP-EP が Create Session Response（PDN1）を MME に転送します。
25	PGW が Modify Bearer Response（PDN2）を S5-GTP-EP に送信します。

ステップ	説明
26	S5-GTP-EP が Modify Bearer Response（PDN2）をサービスポッド（SGW）に転送します。
27	サービスポッド（SGW）が Create Session Response（PDN2）を S11-GTP-EP に送信します。
28	サービスポッド（SGW）が CDL エンドポイントに対する PDN2 セッションを更新します。
29	S11-GTP-EP が Create Session Response（PDN2）を MME に転送します。
30	MME が Delete Session Request（PDN1）[SI=1] を古い SGW に送信します。
31	古い SGW が Delete Session Response で MME に応答します。

サービスポッド（SGW）へのマルチ PDN S1 ハンドオーバー SGW の再配置コールフロー

ここでは、サービスポッド（SGW）へのマルチ PDN S1 ハンドオーバー SGW の再配置コールフローについて説明します。

図 *82 :* サービスポッド（*SGW*）へのマルチ *PDN S1* ハンドオーバー *SGW* の再配置コールフロー

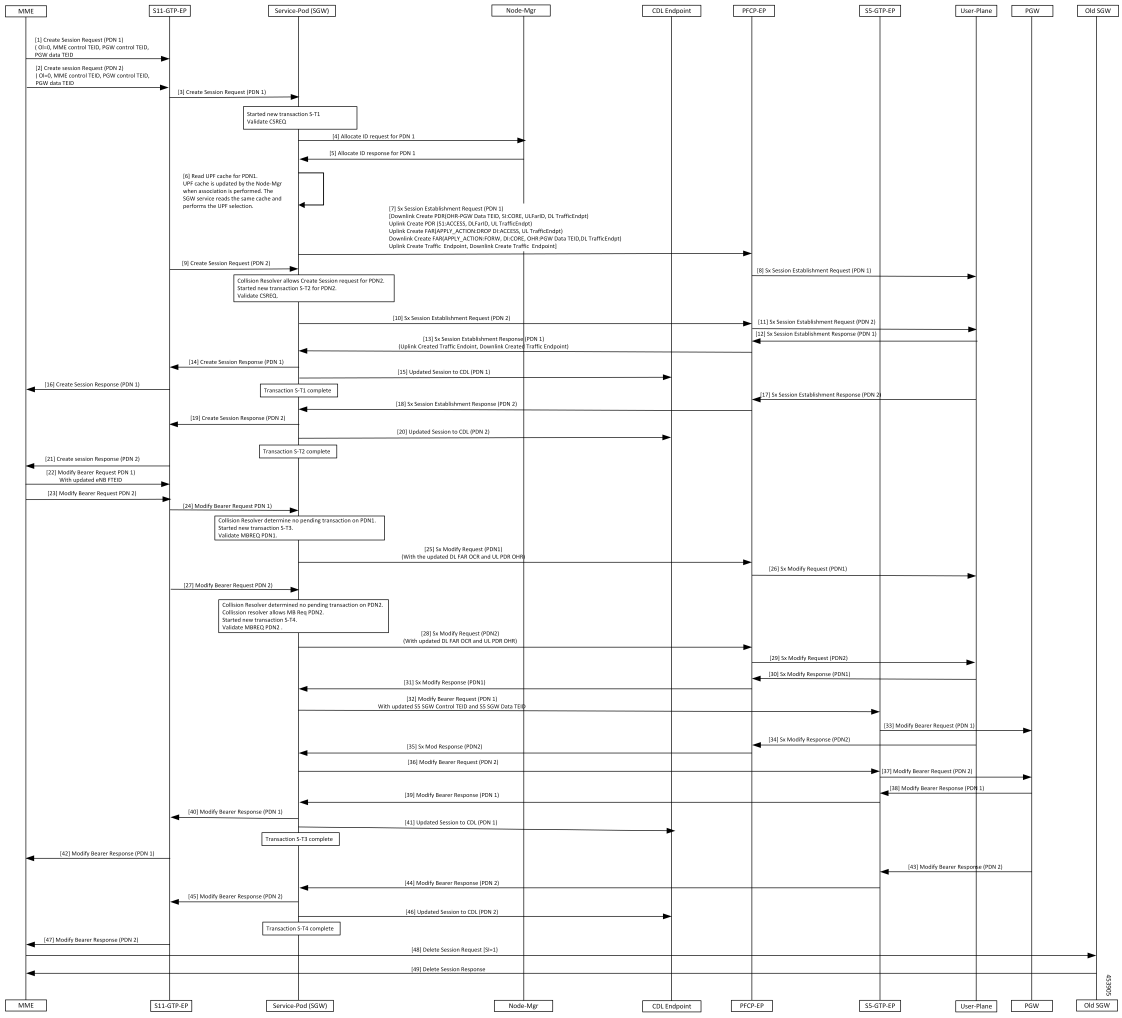


表 *158 :* サービスポッド（*SGW*）へのマルチ *PDN S1* ハンドオーバー *SGW* の再配置コールフローの説明

ステップ	説明
1	MME が、（OI = 0、MME 制御 TEID、PGW 制御 TEID、および PGW データ TEID）とともに、セッション作成要求（PDN1）を S11-GTP-EP に送信します。
2	MME が、（OI = 0、MME 制御 TEID、PGW 制御 TEID、および PGW データ TEID）とともに、セッション作成要求（PDN2）を S11-GTP-EP に送信します。
3	S11-GTP-EP がサービスポッド（SGW）にセッション作成要求（PDN1）を送信します。
4	サービスポッド（SGW）が Node-Mgr に ID の割り当て（PDN1）を要求します。
5	Node-Mgr がサービスポッド（SGW）に ID 割り当て応答（PDN1）で応答します。

ステップ	説明
6	サービスポッド（SGW）が UPF 選択を実行します。
7	サービスポッド（SGW）が PFCP-EP への Sx セッション確立要求（PDN1）をトリガーします。
8	PFCP-EP が Sx セッション確立要求（PDN1）をユーザープレーンに転送します。
9	サービスポッド（SGW）が S11-GTP-EP からセッション作成要求（PDN2）を受信します。
10	サービスポッド（SGW）が Sx セッション確立要求（PDN2）を PFCP-EP に送信します。
11	PFCP-EP が Sx Session Establishment Request（PDN2）をユーザープレーンに転送します。
12	ユーザープレーンが PFCP-EP に対して Sx セッション確立応答（PDN1）で応答します。
13	PFCP-EP が Sx セッション確立応答（PDN1）をサービスポッド（SGW）に転送します。
14	サービスポッド（SGW）がセッション作成応答（PDN1）を S11-GTP-EP に送信します。
15	サービスポッド（SGW）が CDL エンドポイントに対する PDN1 セッションを更新します。
16	S11-GTP-EP がセッション作成応答（PDN1）を MME に転送します。
17	ユーザープレーンが PFCP-EP に対して Sx セッション確立応答（PDN2）で応答します。
18	PFCP-EP が Sx セッション確立応答（PDN2）をサービスポッド（SGW）に転送します。
19	サービスポッド（SGW）がセッション作成応答（PDN2）を S11-GTP-EP に送信します。
20	サービスポッド（SGW）が CDL エンドポイントに対する PDN2 セッションを更新します。
21	S11-GTP-EP がセッション作成応答（PDN2）を MME に転送します。
22	MME が S11-GTP-EP にベアラー変更要求（PDN1）を送信します。
23	MME が S11-GTP-EP にベアラー変更要求（PDN2）を送信します。

ステップ	説明
24	S11-GTP-EP がサービスポッド（SGW）にベアラー変更要求を転送します。
25	サービスポッド（SGW）が PFCP-EP に Sx 変更要求（PDN1）を送信します。
26	PFCP-EP が Sx 変更要求（PDN1）をユーザープレーンに転送します。
27	S11-GTP-EP がサービスポッド（SGW）にベアラー変更要求（PDN2）を送信します。
28	サービスポッド（SGW）が更新された DL FAR OCR と UL PDR OHR を含む Sx 変更要求（PDN2）を PFCP-EP に送信します。
29	PFCP-EP が Sx 変更要求（PDN2）をユーザープレーンに転送します。
30	ユーザープレーンが PFCP-EP に対して Sx 変更要求（PDN1）で応答します。
31	PFCP-EP が Sx 変更要求（PDN1）をサービスポッド（SGW）に転送します。
32	サービスポッド（SGW）が、更新された S5 SGW 制御 TEID と S5 SGW データ TEID とともに、ベアラー変更要求（PDN1）を S5-GTP-EP に送信します。
33	S5-GTP-EP がベアラー変更要求（PDN1）を PGW に転送します。
34	ユーザープレーンが PFCP-EP に Sx 変更応答（PDN2）を送信します。
35	PFCP-EP が Sx 変更応答（PDN2）をサービスポッド（SGW）に転送します。
36	サービスポッド（SGW）が S5-GTP-EP にベアラー変更要求（PDN2）を送信します。
37	S5-GTP-EP がベアラー変更要求（PDN2）を PGW に転送します。
38	PGW が S5-GTP-EP に対してベアラー変更応答（PDN1）で応答します。
39	S5-GTP-EP がベアラー変更応答（PDN1）をサービスポッド（SGW）に転送します。
40	サービスポッド（SGW）がベアラー変更応答（PDN1）を S11-GTP-EP に送信します。
41	サービスポッド（SGW）が CDL エンドポイントに対する PDN1 セッションを更新します。
42	S11-GPT-EP がベアラー変更応答（PDN1）を MME に転送します。
43	PGW が S11-GTP-EP に対してベアラー変更応答（PDN1）で応答します。
44	S5-GTP-EP がベアラー変更応答（PDN2）をサービスポッド（SGW）に転送します。
45	サービスポッド（SGW）がベアラー変更応答（PDN2）を S11-GTP-EP に送信します。

ステップ	説明
46	サービスポッド（SGW）が CDL エンドポイントに対する PDN2 セッションを更新します。
47	S11-GPT-EP がベアラー変更応答（PDN2）を MME に転送します。
48	MME が古い SGW に対してセッション削除要求 [SI=1] をトリガーします。
49	古い SGW がセッションを削除し、セッション削除応答で応答します。

同一 PDN に対する複数の CBR のコールフロー

このセクションでは、同一 PDN に対する複数の CBR のコールフローについて説明します。

図 *83 :* 同一 *PDN* に対する複数の *CBR* のコールフロー

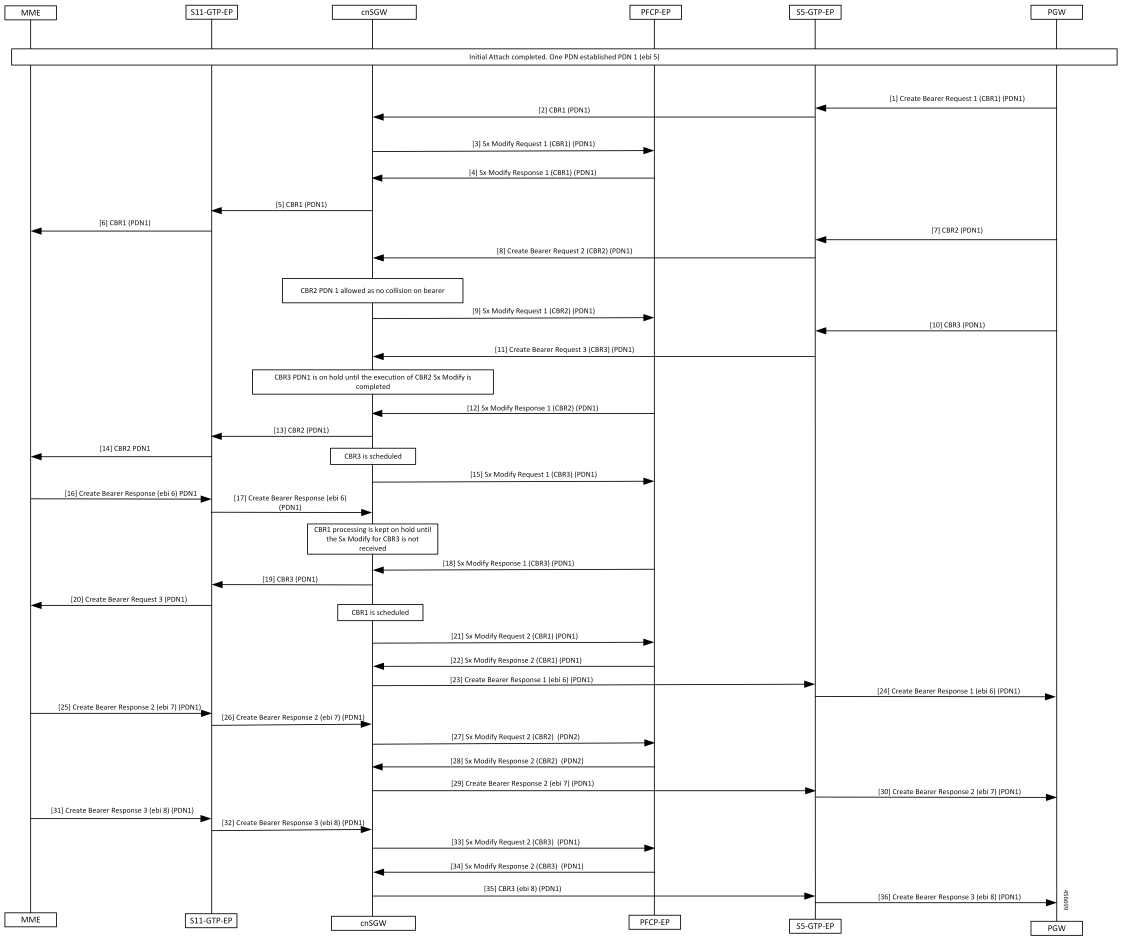


表 *159* : 同一 *PDN* に対する複数の *CBR* のコールフローの説明

ステップ	説明
1	初期接続が完了しました。1 つの PDN が PDN1（ebi 5）を確立しています。 PGW が S5-GTP-EP に PDN1 のベアラー作成要求 1 を送信します。
2	S5-GTP-EP が cnSGW に PDN1 のベアラー作成要求 1 を転送します。
3	cnSGW が PFCP-EP に PDN1 の Sx 変更要求 1 を送信します。
4	cnSGW が PFCP-EP から PDN1 の Sx 変更応答 1 を受信します。
5	cnSGW が S11-GTP-EP に PDN1 のベアラー作成要求 1 を送信します。
6	S11-GTP-EP が MME に PDN1 のベアラー作成要求 1 を送信します。
7	PGW が S5-GTP-EP にベアラー作成要求 2 を送信します。
8	S5-GTP-EP が cnSGW にベアラー作成要求 2 を送信します。
9	cnSGW が PFCP-EP に PDN1 の Sx 変更要求 1（ベアラー作成要求 2）を送信します。
10	PGW が S5-GTP-EP に PDN1 のベアラー作成要求 3 を送信します。
11	S5-GTP-EP が cnSGW に PDN1 のベアラー作成要求 3 を送信します。
12	PFCP-EP が cnSGW に PDN1 の Sx 変更応答 1（CBR2）を送信します。
13	cnSGW が S11-GTP-EP に PDN1 のベアラー作成要求 2 を送信します。
14	S11-GTP-EP が MME に PDN1 のベアラー作成要求 2 を送信します。
15	cnSGW が PFPC-EP に PDN1 の Sx 変更要求 1（CBR3）を送信します。
16	MME が S11-GTP-EP に PDN1 のベアラー作成応答 1（ebi6）を送信します。
17	S11-GTP-EP が cnSGW に PDN1 のベアラー作成応答 1（ebi6）を送信します。
18	cnSGW が PFPC-EP への PDN1 の Sx 変更応答 1（CBR3）を受信します。
19、20	cnSGW が S11-GTP-EP にベアラー作成要求 3 を送信します。
20	S11-GTP-EP が MME にベアラー作成要求 3 を送信します。
21	cnSGW が PFPC-EP に PDN1 の Sx 変更要求 2（CBR1）を送信します。
22	cnSGW が PFPC-EP への PDN1 の Sx 変更応答 2（CBR1）を受信します。
23	cnSGW が S5-GTP-EP に PDN1 のベアラー作成応答 1（ebi 6）を送信します。

ステップ	説明
24	S5-GTP-EP が PGW に PDN1 のベアラー作成応答 1（ebi 6）を送信します。
25	MME は、MME から送信された PDN1 のベアラー作成応答 2（ebi 7）を cnSGW に送信します。
26	S11-GTP-EP は、MME から送信された PDN1 のベアラー作成応答 2（ebi 7）を cnSGW に送信します。
27	cnSGW が PFCP-EP に PDN1 の Sx 変更要求 2（CBR2）を送信します。
28	PFCP-EP が cnSGW に PDN1 の Sx 変更応答 2（CBR2）を送信します。
29	cnSGW が S5-GTP-EP に PDN1 のベアラー作成応答 2（ebi 7）を送信します。
30	S5-GTP-EP が PGW に PDN1 のベアラー作成応答 2（ebi 7）を送信します。
31	MME が S11-GTP-EP に PDN1 のベアラー作成応答 3（ebi 8）を送信します。
32	S11-GTP-EP が cnSGW に PDN1 のベアラー作成応答 3（ebi 8）を送信します。
33	cnSGW が PFPC-EP に PDN1 の Sx 変更要求 2（CBR3）を送信します。
34	PFCP-EP が cnSGW に PDN1 の Sx 変更応答 2（CBR3）を送信します。
35	cnSGW が S5-GTP-EP に PDN1 のベアラー作成応答 3（ebi 8）を送信します。
36	S5-GTP-EP が PGW に PDN1 のベアラー作成応答 3（ebi 8）を送信します。

コリジョンリゾルバの破棄処理コールフロー

ここでは、コリジョンリゾルバの破棄処理コールフローについて説明します。

図 *84 :* コリジョンリゾルバの破棄処理コールフロー

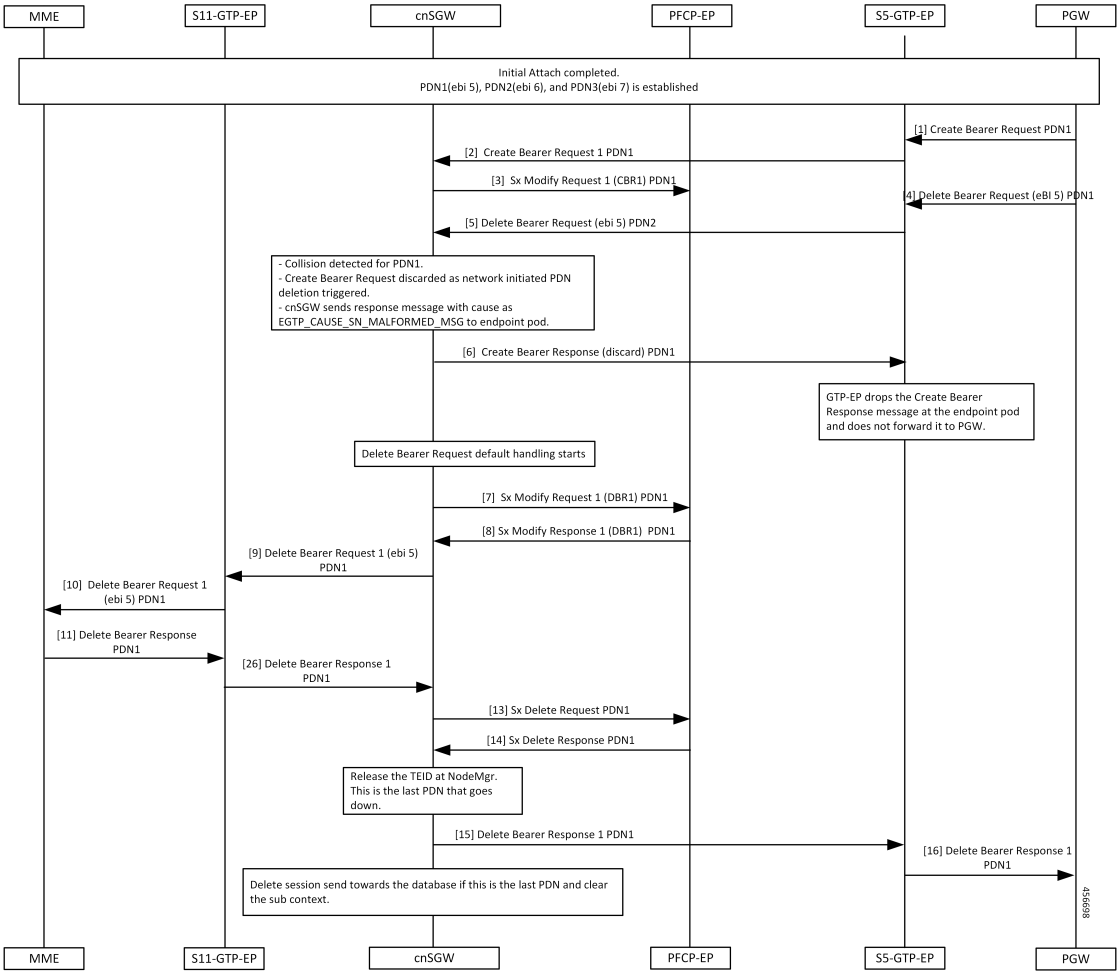


表 *160 :* コリジョンリゾルバの破棄処理コールフローの説明

ステップ	説明
1	初期接続手順が完了しました。 PDN1（ebi 5）、PDN2（ebi 6）、および PDN3（ebi 7）が確立されます。 PGW が PDN1 の Create Bearer Request を S5-GTP-EP に送信します。
2	PFCP-EP が PDN1 の Create Bearer Request 1 を cnSGW に送信します。
3	cnSGW が PDN1 の Sx Modify Request 1 を PFCP-EP に送信します。
4、5	PGW が PDN1 のDelete Bearer Request（ebi5）を cnSGW に送信します。

ステップ	説明
6	PDN1 でコリジョンが検出されました。 ネットワークで開始された PDN の削除要求がトリガーされると、Create Bearer Request は破棄されます。 cnSGW が EGTP_CAUSE_SN_MALFORMED_MSG の理由で応答メッセージをエンドポイントポッドに送信します。 cnSGW が PDN1 の Create Bearer Response（discard 1）を S5-GTP-EP に送信します。
7	cnSGW が PDN1 の Sx Modify Request 1（DBR1）を PFPC-EP に送信します。
8	cnSGW が PFPC-EP から PDN1 の Sx Modify Request 1（DBR1）を受信します。
9	cnSGW が Delete Bearer Request 1 を S11-GTP-EP に送信します。
10	S11-GTP-EP が Delete Bearer Request 1 を MME に送信します。
11	MME が Delete Bearer Response 1 を S11-GTP-EP に送信します。
12	S11-GTP-EP が Delete Bearer Response 1 を cnSGW に送信します。
13	cnSGW が PDN1 の Sx Delete Request を PFCP-EP に送信します。
14	PFPC-EP が PDN1 の Sx Delete Response を cnSGW に送信します。
15	cnSGW が PDN1 の Delete Bearer Response 1 を PGW に送信します。
16	S5-GTP-EP が PDN1 の Delete Bearer Response を PGW に送信します。

一時停止処理コールフロー

このセクションでは、一時停止処理のコールフローについて説明します。

図 *85* : 一時停止処理コールフロー

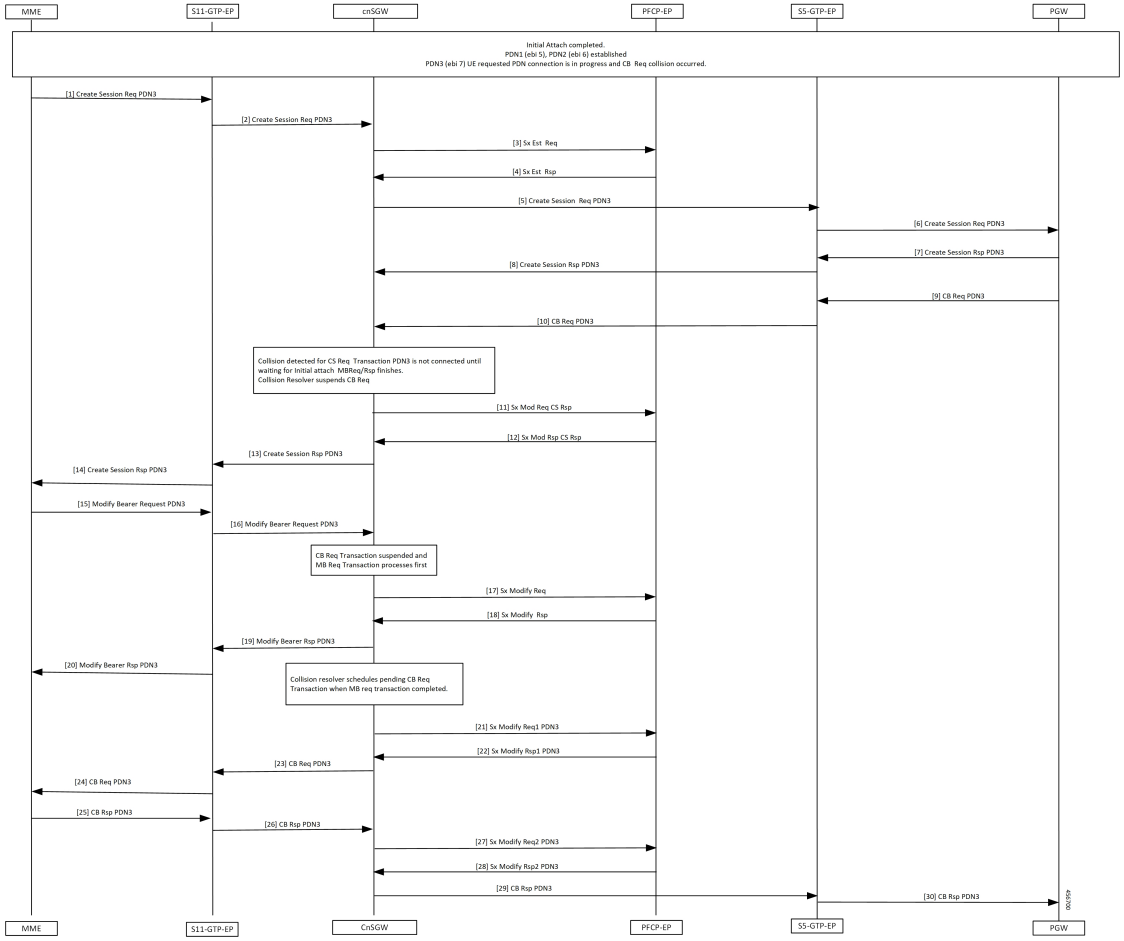


表 *161* : 一時停止処理コールフローの説明

ステップ	説明
1	初期接続プロセスが完了しました。 PDN1（ebi 5）と PDN2（ebi 6）が確立されます。 PDN3（ebi 7）UE は、PDN 接続が進行中であることを要求しました。ベアラー作成要求のコリジョンが発生しました。 MME が PDN3 のセッション作成要求を S11-GTP-EP に送信します。
2	S11-GTP-EP が PDN3 のセッション作成要求を cnSGW に送信します。
3	cnSGW が Sx 確立要求を PFCP-EP に送信します。
4	PFCP-EP が Sx 確立応答を cnSGW に送信します。

ステップ	説明
5	cnSGW が PDN3 のセッション作成要求を PGW に送信します。
6	S5-GTP-EP が PDN3 のセッション作成要求を PGW に送信します。
7	PGW が PDN3 のセッション作成応答を S5-GTP-EP に送信します。
8	S5-GTP-EP が PDN3 のセッション作成応答を cnSGW に送信します。
9	PGW が PDN3 のベアラー作成要求を S5-GTP-EP に送信します。
10	S5-GTP-EP が cnSGW に PDN3 のベアラー作成要求を送信します。
11	PDN3 のセッション作成要求トランザクションで検出されたコリジョンが、初期接続時のベアラー変更要求またはベアラー変更応答の完了を待機しているときに接続されます。 コリジョンリゾルバがベアラー作成要求を一時停止します。 cnSGW は、Sx 変更要求およびセッション作成要求を PFCP-EP に送信します。
12	PFCP-EP が、Sx 変更応答とセッション作成応答を cnSGW に送信します。
13	cnSGW は、S11-GTP-EP に送信された PDN3 のセッション作成応答を送信します。
14	S11-GTP-EP は、MME に送信された PDN3 のセッション作成応答を送信します。
15	MME が S11-GTP-EP に PDN3 のベアラー変更要求を送信します。
16	S11-GTP-EP が cnSGW に PDN3 のベアラー変更要求を送信します。
17	cnSGW が PFCP-EP に PDN3 のベアラー変更要求を送信します。
18	PFCP-EP が Sx 変更応答を cnSGW に送信します。
19	cnSGW が、PDN3 のベアラー変更応答を S11-GTP-EP に送信します。
20	S11-GTP-EP が PDN3 のベアラー変更応答を MME に送信します。
21	cnSGW が PDN3 の Sx 変更要求 1 を PFCP-EP に送信します。
22	PFPC-EP が PFPC-EP に PDN3 の Sx 変更応答 1 を送信します。
23	cnSGW は、S11-GTP-EP に送信された PDN3 のベアラー作成要求を送信します。
24	S11-GTP-EP は、MME に送信された PDN3 のベアラー作成要求を送信します。
25	MME が PDN3 のベアラー作成応答を S11-GTP-EP に送信します。
26	S11-GTP-EP が PDN3 のベアラー作成応答を cnSGW に送信します。

ステップ	説明
27	cnSGW が PDN3 の Sx 変更要求 2 を PFCP-EP に送信します。
28	PFCP-EP が PDN3 の Sx 変更応答 2 を cnSGW に送信します。
29	cnSGW が PDN3 のベアラー作成応答を PGW に送信します。
30	S5-GTP-EP が PGW に PDN3 のベアラー作成応答を送信します。

優先度の低い手順の中止処理

このセクションでは、優先度の低い手順の中止処理のコールフローについて説明します。

図 *86 :* 優先度の低い手順の中止処理のコールフロー

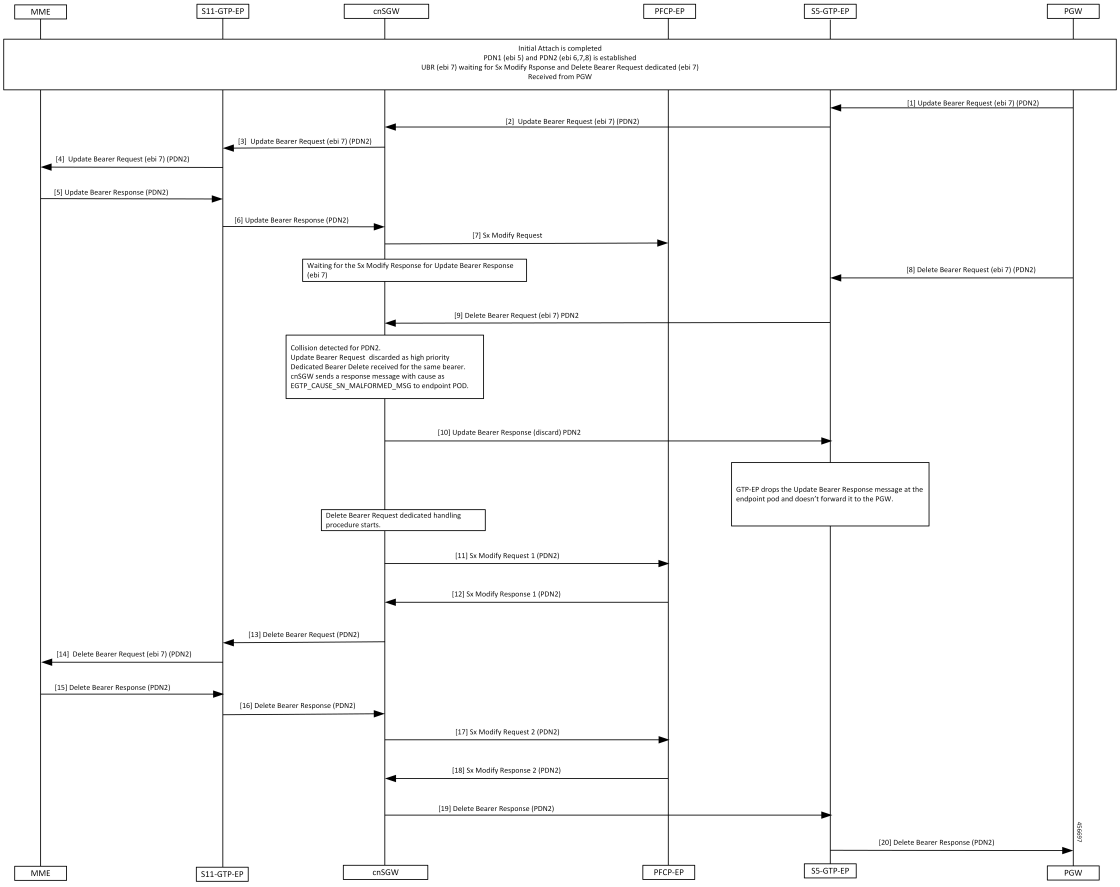


表 *162 :* 優先度の低い手順の中止処理のコールフローの説明

ステップ	説明
1	初期接続手順が完了しました。 PDN1（ebi 5）と PDN2（ebi 6,7,8）が確立されます。 ベアラー更新要求（ebi 7）は、PGW から受信した Sx 変更応答と専用ベアラー要求（ebi 7）を待っています。 PGW が、ベアラー更新要求（eBi 7）PDN2 を S5-GTP-EP に送信します。
2	PFCP-EP が、ベアラー更新要求（eBi 7）PDN2 を cnSGW に送信します。
3	cnSGW が、ベアラー更新要求（eBi 7）PDN2 を S11-GTP-EP に送信します。
4	S5-GTP-EP が、ベアラー更新要求（eBi 7）PDN2 を MME に転送します。
5	MME が、ベアラー更新応答を S11-GTP-EP に送信します。
6	S11-GTP-EP が、ベアラー更新応答を cnSGW に送信します。
7	cnSGW が、Sx 変更要求を PFCP-EP に送信します。
8	PGW が、PDN2 のベアラー削除要求（eBi 7）を S5-GTP-EP に送信します。
9	S5-GTP-EP が、PDN2 のベアラー削除要求（eBi 7）を cnSGW に転送します。
10	PDN2 でコリジョンが検出されます。 同じベアラーに対して高優先度の専用ベアラー削除が受信されたため、ベアラー更新要求は破棄されます。 cnSGW が、EGTP_CAUSE_SN_MALFORMED_MSG の原因で応答メッセージをエンドポイントポッドに送信します。 cnSGW は、PDN2 のベアラー更新要求（Discard）を S5-GTP-EP に送信します。
11	GTP-EP は、エンドポイントポッドでベアラー更新要求のメッセージをドロップし、PGW には転送しません。 cnSGW が PDN2 の Sx 更新要求 1 を PFCP-EP に送信します。
12	PFCP-EP が、PDN2 の Sx 変更応答 1 を cnSGW に送信します。
13	cnSGW が、PDN2 のベアラー削除要求を S11-GTP-EP に送信します。
14	S11-GTP-EP が PDN2 のベアラー削除要求（ebi 7）を MME に送信します。
15	MME が PDN2 のベアラー削除応答を S11-GTP-EP に送信します。
16	S11-GTP-EP が PDN2 のベアラー削除応答を cnSGW に送信します。

ステップ	説明
17	cnSGW が PDN2 の Sx 更新要求を PFCP-EP に送信します。
18	PFCP-EP が、PDN2 の Sx 変更応答を cnSGW に送信します。
19	cnSGW が S5-GTP-EP に PDN2 のベアラー削除応答を送信します。
20	S5-GTP-EP が PDN2 のベアラー削除応答を PGW に送信します。

二重削除最適化コールフロー

ここでは、二重削除最適化コールフローについて説明します。

図 *87* : 二重削除最適化コールフロー

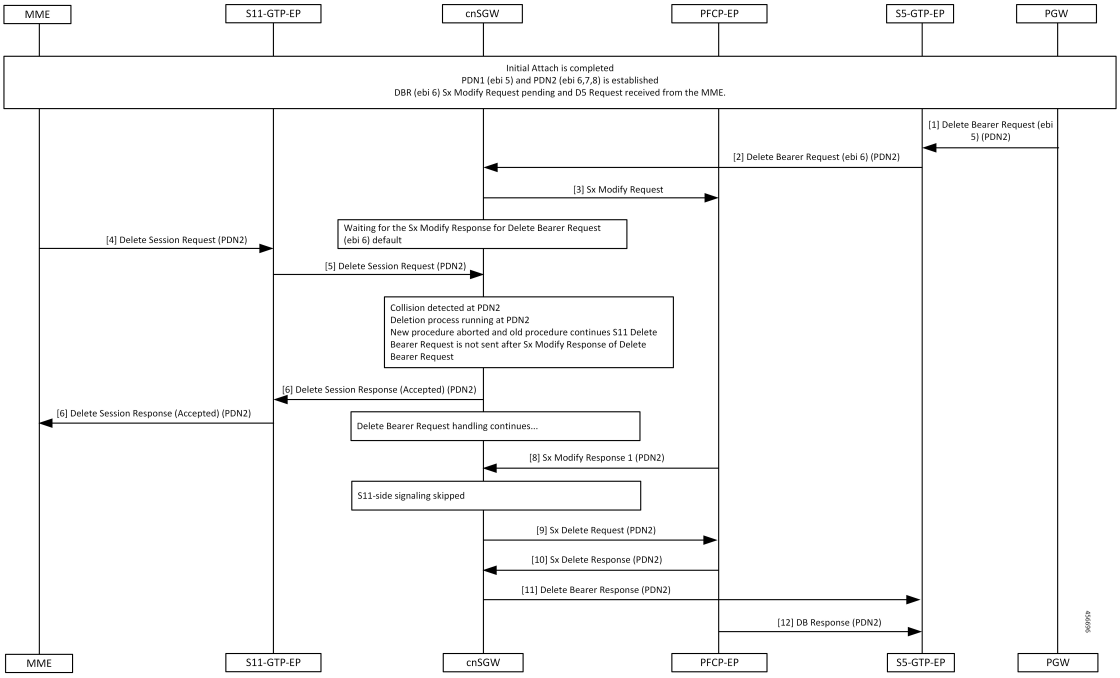
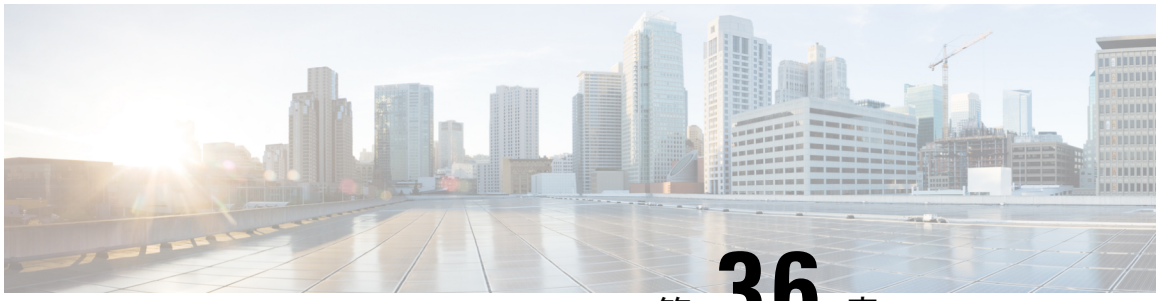


表 *163* : 二重削除最適化コールフローの説明

ステップ	説明
1	初期接続手順が完了しました。 PDN1（ebi 5）と PDN2（ebi 6,7,8）が確立されます。 ベアラー削除要求（ebi 6）と Sx 変更要求が保留中で、セッション削除要求が MME から受信されます。 PGW が S5-GTP-EP に PDN2 のベアラー削除要求（ebi 6）を送信します。

ステップ	説明
2	S5-GTP-EP が cnSGW に PDN2 のベアラー削除要求（ebi 6）を転送します。
3	cnSGW が PFCP-EP に Sx 変更要求を送信します。
4	ベアラー削除要求（ebi 6）デフォルトの Sx 変更応答を待機します。 MME が S11-GTP-EP に PDN2 のセッション削除要求を送信します。
5	S11-GTP-EP が cnSGW に PDN2 のセッション削除要求を転送します。
6	PDN2 でコリジョンが検出されます。 PDN2 で削除プロセスが実行されます。 新しい手順は中止され、古い手順が続行されます。 ベアラー削除要求の Sx 変更応答の後、S11 ベアラー削除要求は送信されません。 cnSGW が S11-GTP-EP に PDN2 のセッション削除応答（承認済み）を送信します。
7	S11-GTP-EP が MME に PDN2 のセッション削除応答（承認済み）を転送します。
8	ベアラー削除要求の処理が進行している間に、PFCP-EP が cnSGW に PDN2 の Sx 変更応答 1 を送信します。
9	cnSGW が PFCP-EP に PDN2 の Sx 削除要求を送信します。
10	PFCP-EP が cnSGW に PDN2 の Sx 削除応答を送信します。
11	cnSGW が S5-GTP-EP に PDN2 のベアラー削除応答を送信します。
12	S5-GTP-EP が PGW に PDN2 のベアラー削除応答を転送します。



第 36 章

ベアラーコマンドサポートの変更と削除

- 機能の概要と変更履歴 （423 ページ）
- 機能説明 （424 ページ）
- 機能の仕組み （424 ページ）

機能の概要と変更履歴

要約データ

表 *164 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	[Enabled]：常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *165 :* マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

cnSGW-C は、Modify Bearer Command（MBC）と Delete Bearer Command（DBC）をサポートしています。

この機能は、SGW サービス、GTP-EP、および UDP プロキシのポッドでサポートされています。

SGW サービスポッドは、次の処理に対応します。

- MBC と DBC
- MBC がトリガーした Update Bearer Request
- DBC がトリガーした Delete Bearer Request

GTPC-EP ポッドは、次の送信を実行します。

- 応答を受信しない場合の Bearer Command Failure Indication（MBCFI）および Delete Bearer Command Failure Indication（DBCFI）。
- Update Bearer Request および Delete Bearer Request をそれぞれ受信した場合の MBCFI と DBCFI（成功）。
- SGW サービスポッドからそれぞれのメッセージを受信した場合の PGW への Update Bearer Response および Delete Bearer Response の返信。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

MBC 障害処理コールフロー

ここでは、MBC 障害処理コールフローについて説明します。

図 *88 : MBC* 障害処理コールフロー

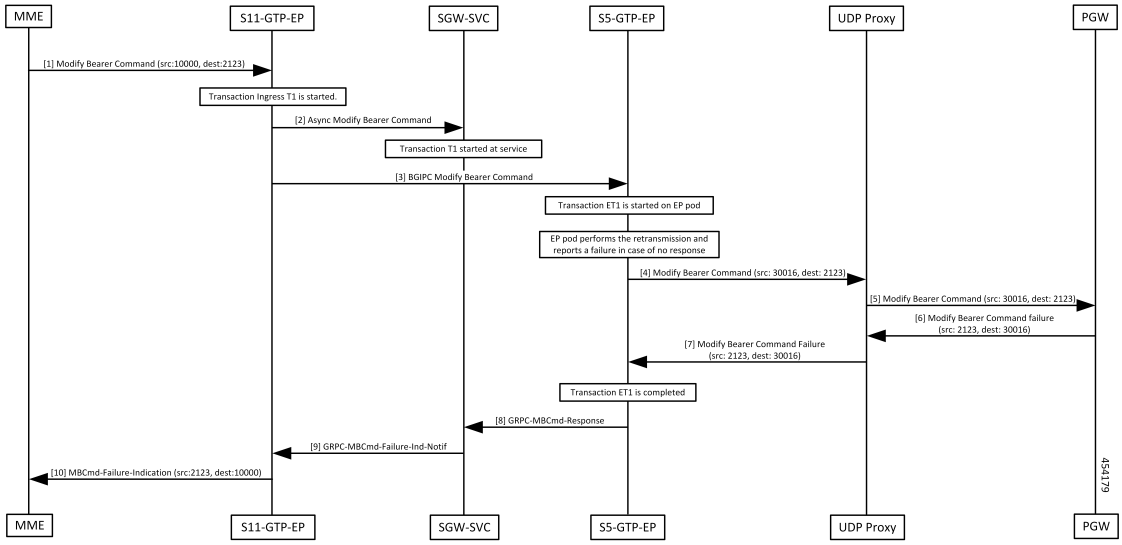


表 *166 : MBC* 障害処理コールフローの説明

ステップ	説明
1	MME が Modify Bearer Command を S11 GTPC-EP ポッドに送信します。
2	GTPC-EP ポッドが ASYNC Modify Bearer Command を SGW-SVC ポッドに送信します。
3	SGW-SVC ポッドが Modify Bearer Command を PGW に転送します。応答のために MBC_info を PDN に保存します。
4	GTPC-EP ポッドが再送信を実行します。応答がない場合、ポッドが Modify Bearer Command Failure Indication をSGW-SVC ポッドに送信します。
5	UDP プロキシが Modify Bearer Command 要求を PGW に送信します。
6	MBCFI が S5 GTPC-EP ポッドで受信され、SGW-SVC ポッドに転送されます。
7	UDP プロキシが Modify Bearer Command の失敗の詳細を S5-GTP-EP に送信します。
8	S5-GTP-EP が GRPC Modify Bearer Command Response を SGW-SVC に送信します。
9	SGW-SVC が GRPC Modify Bearer Command の失敗通知を S11-GTP-EP に送信します。
10	SGW-SVC ポッドが MBCFI を処理し、保存された MBC_Info を含む応答を MME に転送します。

MBC 成功処理コールフロー

ここでは、MBC 成功処理コールフローについて説明します。

図 *89 : MBC* 成功処理コールフロー

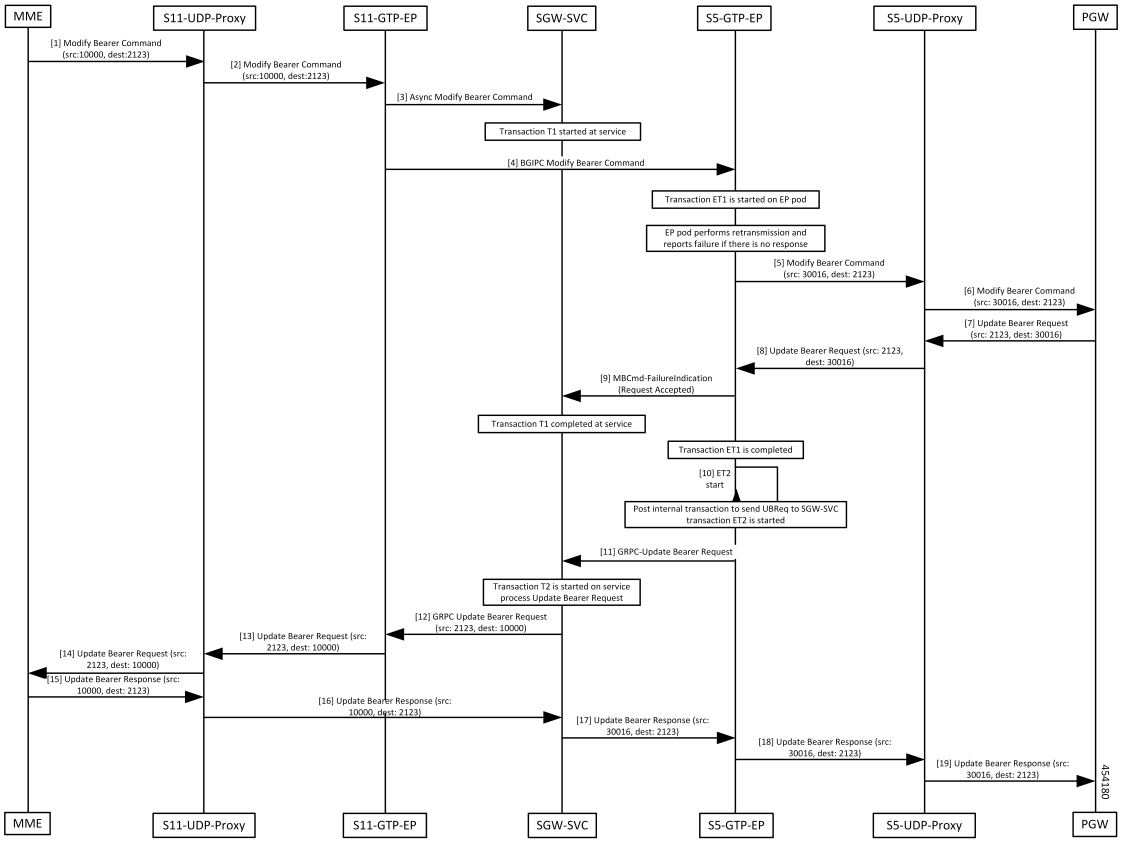


表 *167 : MBC* 成功処理コールフローの説明

ステップ	説明
1	MME がベアラー変更コマンドを S11-UDP-Proxy に送信します。
2	S11-UDP-Proxy がベアラー変更コマンドを S11-GTP-EP に転送します。
3	S11-GTP-EP ポッドが非同期ベアラー変更コマンドを SGW-SVC ポッドに送信します。 SGW-SVC ポッドがベアラー変更コマンドを PGW に転送します。応答用に PDN に MBC_info を保存します。
4	S11-GTP-EP ポッドが再送信を実行します。応答がない場合、ポッドはベアラー変更コマンド障害表示（MBCFI）を S5-GTP-EP ポッドに送信します。
5	S5-GTP-EP がベアラー削除コマンドを S5-UDP-Proxy に送信します。

ステップ	説明
6	S5-UDP-Proxy がベアラー変更コマンドを PGW に転送します。
7	PGW が S5-UDP-Proxy にベアラー更新要求を送信します。
8	S5-UDP-Proxy がベアラー更新要求（送信元：2123、宛先：30016）を S5-GTP-EP に送信します。
9	S5-GTP-EP ポッドが MBCmd-FailureIndication を SGW-SVC ポッドに送信し、トランザクションを終了します。 内部トランザクションの後、GRPC ベアラー更新要求が SGW-SVC ポッドに送信されます。
10	S5-GTP-EP が ET2 を開始します。
11	S5-GTP-EP が GRPC ベアラー更新要求を SGW-SVC ポッドに送信します。
12	SGW-SVC ポッドがベアラー更新要求を処理し、保存された MBC_Info を消費して S11-GTP-EP にベアラー更新要求を送信します。
13	S11-GTP-EP が S11-UDP-Proxy にベアラー更新要求を送信します。
14	S11-UDP-Proxy がベアラー更新要求を MME に転送します。
15	MME が SGW-SVC ポッドにベアラー更新応答を送信します。
16	S11-UDP-Proxy がベアラー更新応答を処理し、SGW-SVC ポッドに送信します。
17	SGW-SVC ポッドがベアラー更新応答を S5-GTP-EP ポッドに転送します。
18	S5-GTP-EP ポッドがベアラー削除応答を S5-UDP-Proxy に送信します。
19	S5-UDP-Proxy がベアラー削除応答を PGW に送信します。

DBC 障害処理コールフロー

このセクションでは、DBC 障害処理のコールフローについて説明します。

図 *90 : DBC* 障害処理コールフロー

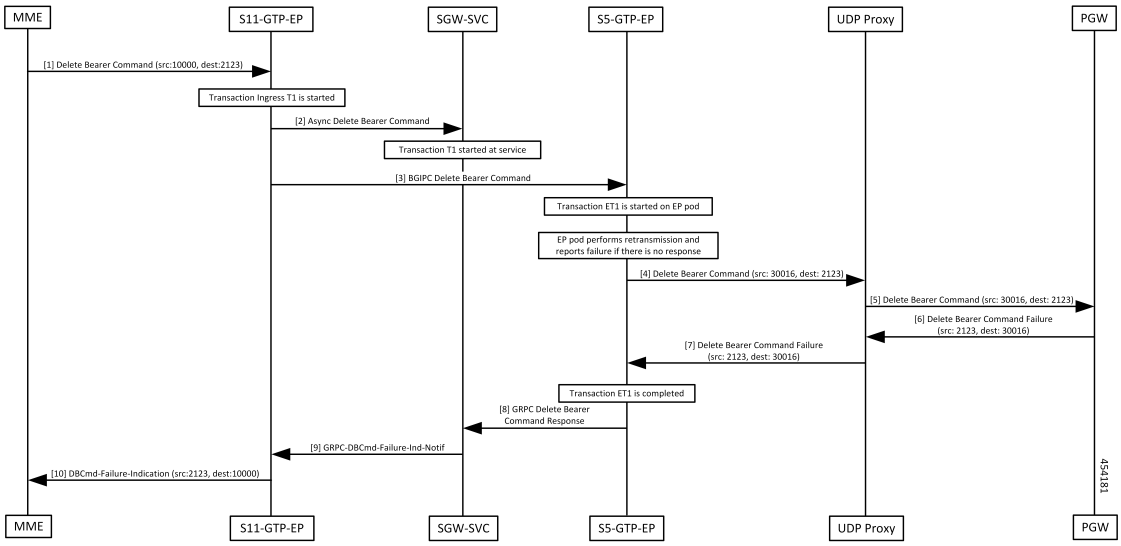


表 *168 : DBC* 障害処理コールフローの説明

ステップ	説明
1	MME が S11-GTP-EP ポッドにベアラー削除コマンドを送信します。
2	S11-GTP-EP ポッドが、ASYNC ベアラー削除コマンドを SGW-SVC ポッドに送信します。
3	SGW-SVC ポッドが、ベアラー削除コマンドを PGW に転送します。応答のために PDN に DBC_info を保存します。 EP ポッドが再送信を実行します。応答がない場合、ポッドはベアラー削除コマンドの失敗表示（DBCFI）を SGW-SVC ポッドに送信します。
4	S5-GTP-EP が、UDP プロキシにベアラー削除コマンドを送信します。
5	UDP プロキシが、ベアラー削除コマンドを PGW に転送します。
6	PGW が UDP プロキシにベアラーベアラー削除コマンドの失敗を送信します。 DBFCI は S5 GTPC-EP ポッドで受信され、SGW-SVC ポッドに転送されます。
7	UDP プロキシが、ベアラー削除コマンドの失敗を S5-GTP-EP に転送します。
8	S5-GTP-EP が、GRPC ベアラー削除コマンド応答を SGW-SVC に送信します。
9	SGW-SVC ポッドは、GRPC-DBCmd-Failure-Ind-Notif を S11-GTP-EP に送信します。
10	S11-GTP-EP ポッドが DBCFI を処理し、保存されている DBC_Info を使用して MME に応答を転送します。

DBC 成功処理コールフロー

ここでは、DBC 成功処理コールフローについて説明します。

図 *91 : DBC* 成功処理コールフロー

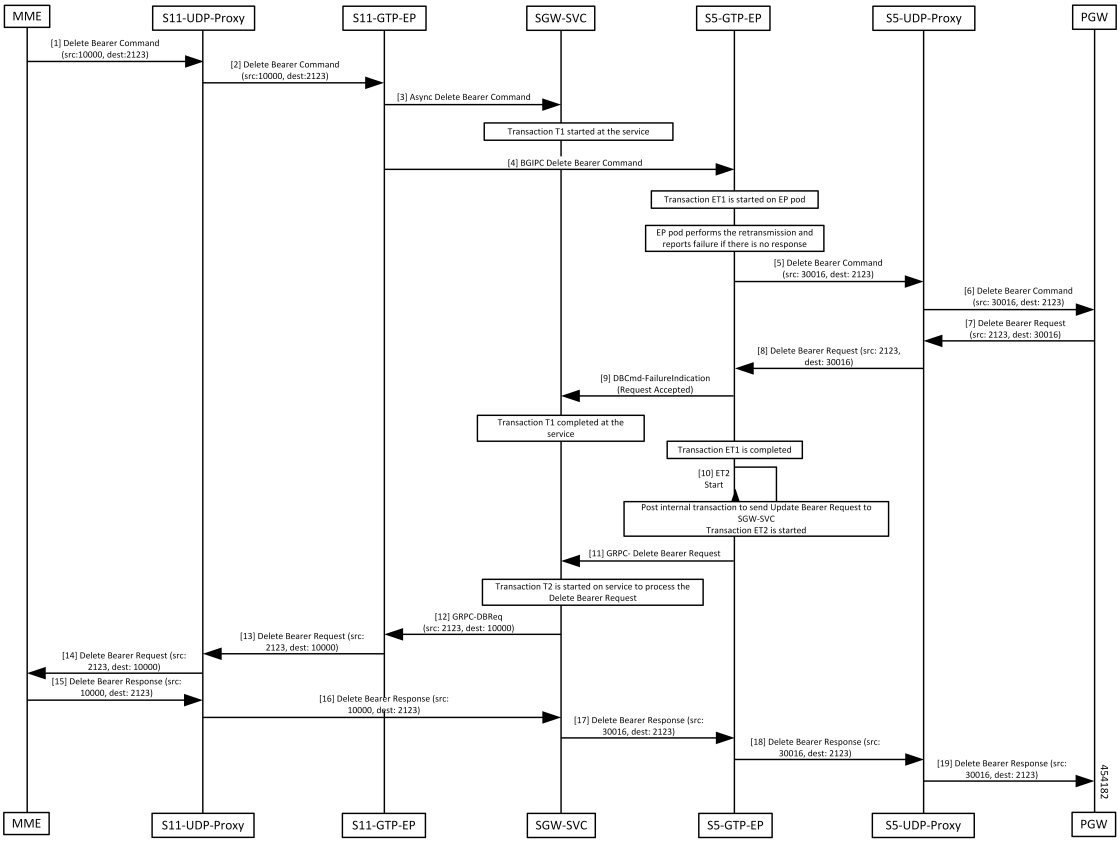
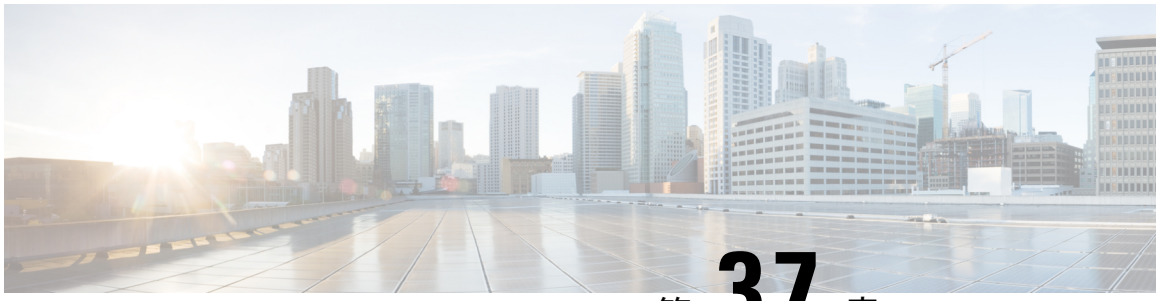


表 *169 : DBC* 成功処理コールフローの説明

ステップ	説明
1	MME が Delete Bearer Command を S11-UDP-Proxy に送信します。
2	S11-UDP-Proxy が Delete Bearer Command を S11-GTP-EP に転送します。
3	S11-UDP-Proxy が ASYNC Delete Bearer Command を SGW-SVC ポッドに送信します。
4	SGW-SVC ポッドが BGIPC Delete Bearer Command を S5-GTP-EP に送信します。応答用の PDN に DBC_info を保存します。
5	EP ポッドが再送信を実行し、応答がない場合は失敗を報告します。ポッドが Delete Bearer Command Failure Indication（DBCFI）を SGW-SVC ポッドに送信します。 S5-GTP-EP が Delete Bearer Command を S5-UDP-Proxy に送信します。

ステップ	説明
6	S5-UDP-Proxy が Delete Bearer Command を PGW に送信します。
7	PGW が Delete Bearer Request を S5-UDP-Proxy ポッドに送信します。
8	S5-UDP-Proxy ポッドが Delete Bearer Request を S5-GTP-EP に転送します。
9	S5-GTP-EP ポッドが DBCFI（要求は ACCEPTED）を SGW-SVC ポッドに送信して、トランザクションが終了します。SGW-SVC ポッドがトランザクションを終了し、この DBFI を消費します。 内部トランザクション後、GRPCE_DBReq が SGW-SVC ポッドに送信されます。
10	ET1 トランザクションが完了すると、S5-GTP-EP が開始します。
11	S5-GTP-EP ポッドが GRPC-DBRequest を SGW-SVC に送信します。
12	SGW-SVC ポッドが Delete Bearer Request を処理し、保存された DBC_Info を使用して、Updated Bearer Request を MME に送信します。
13	S11-GTP-EP ポッドが Delete Bearer Request を S11-UDP-Proxy ポッドに送信します。
14	S11-UDP-Proxy が Delete Bearer Request を MME に転送します。
15	MME が Delete Bearer Response を S11-UDP-Proxy ポッドに送信します。
16	S11-UDP-Proxy が SGW-SVC に対する Delete Bearer Request を処理します。
17	SGW-SVC が Delete Bearer Response を S5-GTP-EP に転送します。
18	S5-GTP-EP ポッドが Delete Bearer Response を S5-UDP-Proxy に送信します。
19	S5-UDP-Proxy が Delete Bearer Response を PGW に送信します。



第 37 章

ベアラー変更要求サポート

- 機能の概要と変更履歴 （431 ページ）
- 機能説明 （431 ページ）
- 機能の仕組み （432 ページ）

機能の概要と変更履歴

要約データ

表 *170*: 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2020.03.0

機能説明

cnSGW-C は、MME からの MBR サービスリクエストをサポートし、UE 状態を IDLE から ACTIVE に変更します。cnSGW-C は、次のサービスリクエストをサポートしています。

- PGW インタラクションを使用しない UE トリガーのサービスリクエスト

- PGW のインタラクションを使用した UE トリガーのサービスリクエスト

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-Cは、UEによってトリガーされたサービス要求の処理中に次のアクションを実行します。

- Sx 変更要求メッセージを UPF に送信します。
  - ダウンリンクの転送アクションルール（FAR）を転送としてマークします。
  - UPF がダウンリンクパケットを eNodeB に送信するように S1 の eNodeB-F TEID 情報を更新します。
- UPF から Sx 変更応答メッセージを受信した後、cnSGW-C は次を実行します。
  - ベアラー変更応答メッセージを MME を送信します。
  - ユーザーロケーション情報（ULI）またはUEタイムゾーンを確認します。タイムゾーンが変更された場合は、ベアラー変更要求を PGW に送信して TAI を更新します。PGW インタラクション要求を含む UE トリガーのサービス要求では、URI または UE タイムゾーンチェックのみを考慮します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

PGW インタラクションを伴わない UE トリガーサービス要求のコールフロー

ここでは、PGW インタラクションを伴わない UE トリガーサービス要求のコールフローについて説明します。

図 *92 : PGW* インタラクションを伴わない *UE* トリガーサービス要求のコールフロー

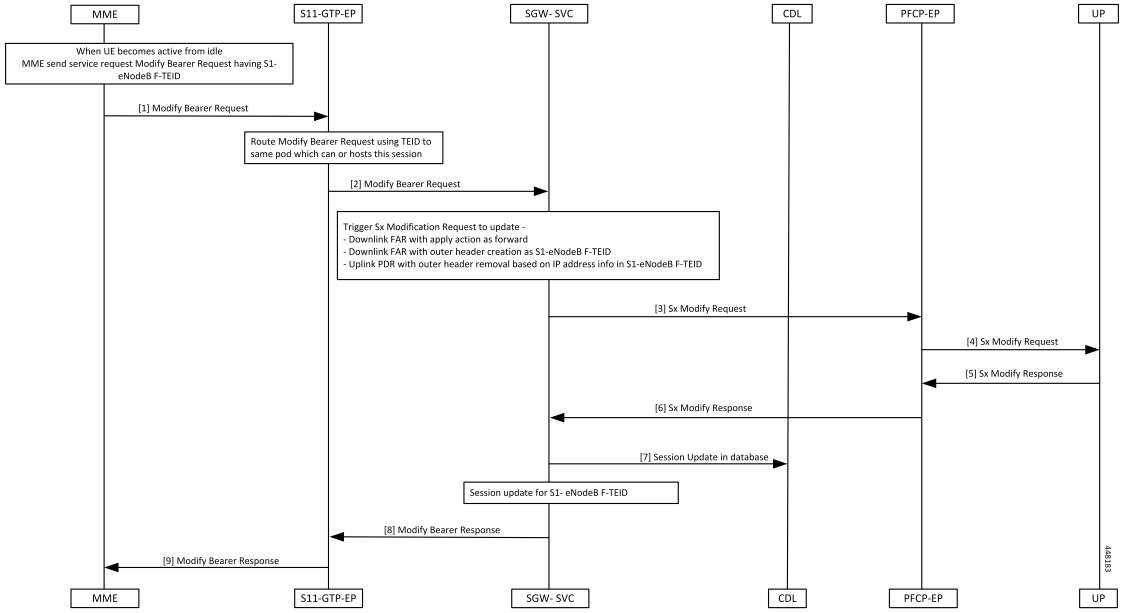


表 *171 : PGW* インタラクションを伴わない *UE* トリガーサービス要求のコールフローの説明

ステップ	説明
1	UE が IDLE 状態から ACTIVE 状態に変わると、MME が s1-eNodeB F-TEID を使用してベアラー変更要求メッセージを cnSGW-C に送信します。
2	S11-GTP-EP が UDP メッセージを復号し、gRPC メッセージに変換します。この gRPC メッセージは、TEID を使用して UE セッションを処理する SGW-SVC ポッドに送信されます。
3	SGW-SVC ポッドが、ローカル入力 TEID を使用してサブスクライバコンテキスト情報を検索します。 SGW-SVC ポッドが、ベアラー変更要求コンテンツを送信し、Sx 変更要求を PFCP-EP に送信します。
4	PFCP-EP が、UDP プロキシを介して UPF に Sx 変更要求メッセージを送信します。
5	UPF が Sx 変更要求メッセージを処理し、Sx 変更応答メッセージを PFCP-EP に送信します。
6	PFCP-EP が、SGW-SVC ポッドに Sx 変更応答メッセージを送信します。
7	SGW-SVC ポッドが、PDN を CONNECTED 状態に変更し、セッションの更新を CDL に送信します。 CDL モジュールはデータベース内の情報を更新します。

ステップ	説明
8	SGW-SVC ポッドが、ベアラー変更応答メッセージを S11-GTP-EP に送信します。
9	S11-GTP-EP が、ベアラー変更応答メッセージを MME に送信します。 MME がベアラー変更応答メッセージを処理します。

PGW インタラクションを伴う UE トリガーサービス要求コールフロー

ここでは、PGW インタラクションを伴う UE トリガーサービス要求コールフローについて説明します。

図 *93 : PGW* インタラクションを伴う *UE* トリガーサービス要求コールフロー

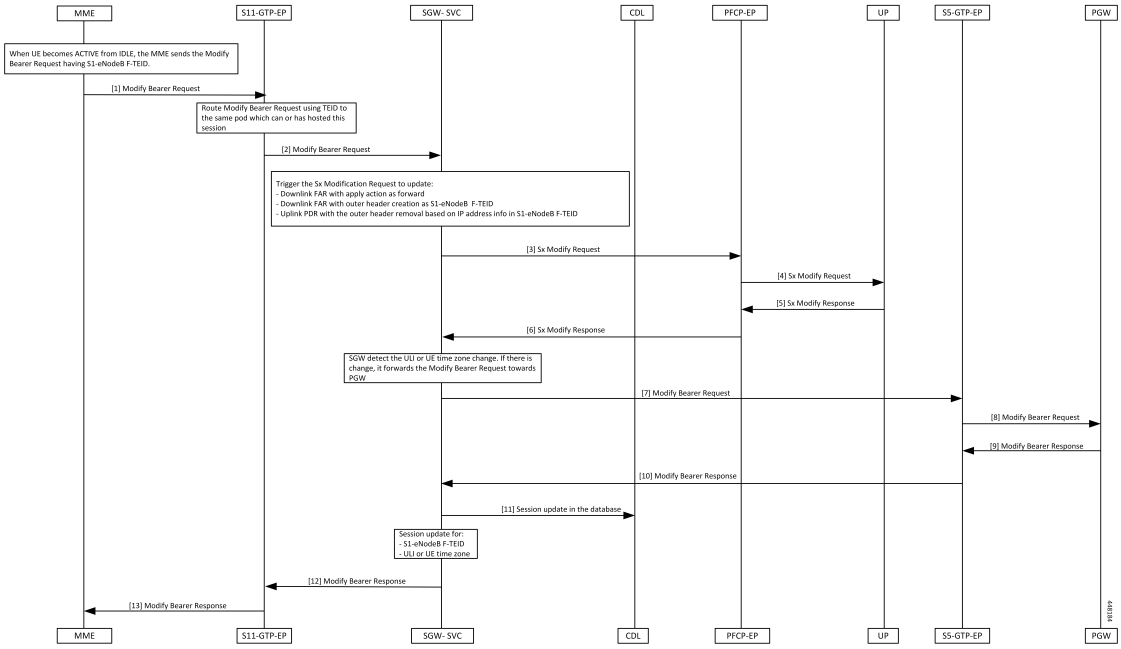
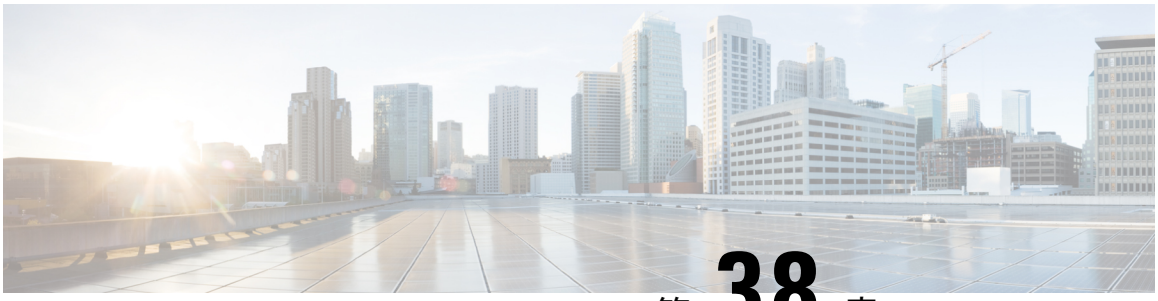


表 *172 : PGW* インタラクションを伴う *UE* トリガーサービス要求コールフローの説明

ステップ	説明
1	UE が IDLE 状態から ACTIVE 状態に変わると、MME が s1-eNodeB F-TEID とともにベアラー変更要求を cnSGW-C に送信します。
2	S11-GTP-EP が UDP メッセージを復号し、gRPC メッセージに変換します。この gRPC メッセージは、TEID を使用して UE セッションを処理する SGW サービスポッドに送信されます。

ステップ	説明
3	SGW サービスポッドが、ローカル入力 TEID を使用してサブスクライバコンテキスト情報を検索します。 ベアラー変更要求の内容を検証し、Sx 変更要求を PFCP-EP に送信します。
4	PFCP-EP が UDP プロキシを介して UPF に Sx 変更要求を送信します。
5	UPF1 が Sx 変更要求を処理し、Sx 変更応答メッセージを送信します。
6	PFCP-EP が SGW サービスポッドに Sx 変更応答メッセージを送信します。
7	SGW サービスポッドが ULI または UE タイムゾーンの変更を検出し、ベアラー変更要求メッセージを S5-GTP-EP に送信します。
8	S5-GTP-EP が PGW にベアラー変更要求メッセージを送信します。
9	PGW がベアラー変更要求メッセージを処理し、ベアラー変更応答メッセージを送信します。
10	S5-GTP-EP が SGW サービスポッドにベアラー変更応答メッセージを送信します。
11	SGW サービスポッドが PDN を CONNECTED 状態に移行し、更新情報を CDL に送信します。 CDL モジュールはデータベース内の情報を更新します。
12	SGW サービスポッドが S11-GTP-EP にベアラー変更応答メッセージを送信します。
13	S11-GTP-EP が MME にベアラー変更応答メッセージを送信します。 MME がベアラー変更応答メッセージを処理します。



第 38 章

モニターサブスクライバおよびプロトコルサポート

- 機能の概要と変更履歴 （437 ページ）
- 機能説明 （438 ページ）
- 機能設定 （438 ページ）

機能の概要と変更履歴

要約データ

表 *173* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	[Enabled]：無効にするには設定が必要
関連資料	該当なし

マニュアルの変更履歴

表 *174* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

cnSGW-C サービスは、SGW サービスパラメータのサブスクライバマップとオペレータポリシーの設定をサポートしています。

機能設定

この機能の設定には、次の手順が含まれます。

- 指定したサブスクライバに関連するメッセージをトレースするようにモニターサブスクライバユーティリティを構成します。詳細については、モニターサブスクライバの設定 （438 ページ） を参照してください。
- モニター サブスクライバ ユーティリティの例。詳細については、設定例 （439 ページ） を参照してください。
- 指定したインターフェイスのパケットをキャプチャするようにモニター プロトコル ユーティリティを構成します。詳細については、モニタープロトコルの設定 （461 ページ） を参照してください。
- モニター プロトコル ユーティリティの例。詳細については、設定例 （462 ページ） を参照してください。
- トランザクションログの要求応答メッセージを構成します。詳細については、トランザクションメッセージの設定 （473 ページ） を参照してください。
- トランザクションログの例。詳細については、設定例 （473 ページ） を参照してください。
- モニターサブスクライバログとプロトコルログにアクセスします。詳細については、ログへのアクセス （478 ページ）を参照してください。

モニターサブスクライバの設定

この機能を設定するには、次の設定を使用します。

```
exec
   monitor subscriber
      capture-duration capture_duration
      dump filename filename_value
      gr-instance gr_instance
      imei imei_value
      imsi imsi_value
      supi supi_value
      list subscriber_list
      internal-messages [ Yes | No ]
      transaction-logs [ Yes | No ]
```

```
nf-service nf_service
end
```

（注） 2021.02 以降のリリースでは、namespace キーワードは廃止され、nf-service に置き換えられています。

注：

- **capture-durationi** *capture_duration*：モニターサブスクライバ機能を有効にする期間を秒単位で指定します。デフォルト値は 300 秒です。
- **supi** *supi_value*：サブスクライバ ID を指定します。例：imsi-123456789 および imsi-123*
- **imsi** *imsi_value*：IMSI 値を指定します。例：123456789 および *
- **imei** *IMEI_value*：IMEI 値を指定します。例：123456789012345 および *
- **internal-messages [Yes | No]**：内部メッセージングを設定します。yes に設定すると、内部メッセージングが有効になります。デフォルトでは、この設定は無効になっています。
- **transaction-logs [Yes | No]** ：トランザクションロギングを設定します。デフォルトでは、この設定は無効になっています。

（注） 任意の時点で、内部メッセージまたはトランザクションログが表示されます。

- **nf-service** *nf_service*：NF サービスを指定します。承認されているサービスは sgw と smf です。デフォルト値は none です。
- **gr-instance** *gr_instance*：cnSGW-C がサブスクライバをモニターする GR インスタンスを指定します。

設定例

次に、設定例を示します。

```
monitor subscriber imsi 123456789 capture-duration 100 internal-messages yes
monitor subscriber imsi 123456789 capture-duration 100 transaction-logs yes
```

出力の例

次に、出力例を示します。

```
monitor subscriber imsi * namespace sgw
supi: imsi-*
captureDuraiton: 300
enableInternalMsg: false
enableTxnLog: false
namespace(deprecated. Use nf-service instead.): sgw
nf-service: none
```

```
gr-instance: 0
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   277  100    89  100   188   5235  11058 --:--:-- --:--:-- --:--:-- 16294
Command: --header Content-type:application/json --request POST --data
{"commandname":"mon_sub","parameters":{"supi":"imsi-*","duration":300,
"enableTxnLog":false,
"enableInternalMsg":false,"action":"start","namespace":"sgw","nf-service":"none",
"grInstance":0}}
http://oam-pod:8879/commands
Result start mon_sub, fileName ->
logs/monsublogs/sgw.imsi-*_TS_2021-08-15T12:36:17.569800845.txt
Starting to tail the monsub messages from file:
logs/monsublogs/sgw.imsi-*_TS_2021-08-15T12:36:17.569800845.txt
Defaulting container name to oam-pod.
Use 'kubectl describe pod/oam-pod-0 -n cn' to see all of the containers in this pod.
      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.288997
      Message: Sx Session Establishment Request
      Description: Sx Session Establishment Request Message from SGWC to SGWU
      Source: 209.165.201.19 (SGW.protocol.DC.Local.1)
      Destination: 209.165.201.20 (SGW.udp-proxy.DC.Local.0)
      PAYLOAD:
        Sx Session Establishment Request:
            Sx Session Establishment Request:
                FSeid:
                    Seid: 1297038098512740679
                    IPv4Address: 209.165.201.19
                CreatePdr:
                    CreatePdr[0]:
                        PdrId: 1
                        Precedence: 0
                        Pdi:
                            SrcIf: CORE
                            UeIp:
                                Src: false
                                Dst: false
                                IPv4Addr: 209.165.201.30
                            TEndpointId: 1
                            Valid: true
                        OuterHdrRem: 0
                        FarId:
                            FarId[0]: 1
                        Qfi: 0
                        OuterHdrRemValid: false
                    CreatePdr[1]:
                        PdrId: 2
                        Precedence: 0
                        Pdi:
                            SrcIf: ACCESS
                            UeIp:
                                Src: false
                                Dst: false
                                IPv4Addr: 209.165.201.30
                            TEndpointId: 2
                            Valid: true
                        OuterHdrRem: 0
                        FarId:
                            FarId[0]: 2
                        Qfi: 0
                        OuterHdrRemValid: false
                CreateFar:
                    CreateFar[0]:
                        FarId: 1
```

```
        ApplyAction:
            Drop: true
            Frwd: false
            Buff: false
            Nocp: false
            Dupl: false
            Valid: true
        FwdParams:
            DestIf: ACCESS
            RedirectInfo:
                AddrType: 0
                Valid: false
            OuterHdr:
                OuterHdrDesc: 0
                Teid: 0
                IPv4Address: 209.165.201.30
                Port: 0
                Valid: false
            TEndptId: 2
            OuterPktTos: 255
            InnerPktTos: 255
            TosOpt:
                CopyInner: false
                CopyOuter: false
            SendTos: 0
            PfcpSmFlags:
                Drobu: false
                Qaurr: false
                Sndem: false
                Valid: false
            Valid: true
            NextHopId: 0
        DuplParams:
            DestIf: ACCESS
            OuterHdr:
                OuterHdrDesc: 0
                Teid: 0
                IPv4Address: 209.165.201.30
                Port: 0
                Valid: false
            InterceptInfo:
                InterceptId: 0
                ChargingId: 0
                SmfLiNodeId:
                    IpDesc: 0
                    IPv4Address: 209.165.201.30
                    Valid: false
                PduSessionId: 0
                Valid: false
            Valid: false
        BarId: 0
    CreateFar[1]:
        FarId: 2
        ApplyAction:
            Drop: true
            Frwd: false
            Buff: false
            Nocp: false
            Dupl: false
            Valid: true
        FwdParams:
            DestIf: CORE
            RedirectInfo:
                AddrType: 0
```

```
                    Valid: false
                OuterHdr:
                    OuterHdrDesc: 0
                    Teid: 0
                    IPv4Address: 209.165.201.30
                    Port: 0
                    Valid: false
                TEndptId: 1
                OuterPktTos: 255
                InnerPktTos: 255
                TosOpt:
                    CopyInner: false
                    CopyOuter: false
                SendTos: 0
                PfcpSmFlags:
                    Drobu: false
                    Qaurr: false
                    Sndem: false
                    Valid: false
                Valid: true
                NextHopId: 0
            DuplParams:
                DestIf: ACCESS
                OuterHdr:
                    OuterHdrDesc: 0
                    Teid: 0
                    IPv4Address: 209.165.201.30
                    Port: 0
                    Valid: false
                InterceptInfo:
                    InterceptId: 0
                    ChargingId: 0
                    SmfLiNodeId:
                        IpDesc: 0
                        IPv4Address: 209.165.201.30
                        Valid: false
                    PduSessionId: 0
                    Valid: false
                Valid: false
            BarId: 0
    CreateTEndpt:
        CreateTEndpt[0]:
            EndpointId: 1
            FTeid:
                Teid: 0
                IPv4Address: 209.165.201.30
                ChooseId: 0
            BearerLvlInfo:
                Valid: 1
                Qci: 6
        CreateTEndpt[1]:
            EndpointId: 2
            FTeid:
                Teid: 0
                IPv4Address: 209.165.201.30
                ChooseId: 0
            BearerLvlInfo:
                Valid: 1
                Qci: 6
    PdnType: 0
    UplaneInacTimer: 0
    MetaData: From:209.165.201.19:10665->To:209.165.201.20:8805
    Supi:
    Seid: 1297038098512740679
```

```
                    Seqno: 4252
                    Version: 0
                    MsgPriority: false
                    MsgPriorityVal: 0
                    Cmnid: 0
                    Rseid: 0
                    IntfType: 0
                    HdrLen: 0
                    MsgLen: 0
                    UserIDInfo:
                        Imsi: 123456789012348
                        Imei: 123456786666660
                        Msisdn: 223310101010101
                        Valid: true
                    XHeaderInfo:
                        RatType:
                        Valid: false
                    CfPolicyId:
                        PolicyId: 0
                        Valid: false
                    ChargingDisabled:
                        Valid: false
                        Value: false
                    ChargingParams:
                        Valid: 0
                        GyOfflineChargingEnabled: 0
                    NextHopIPv4: 0

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.371114
      Message: Sx Session Establishment Response
      Description: Sx Session Establishment Response Message from SGWU to SGWC
      Source: 209.165.201.20 (SGW.udp-proxy.DC.Local.0)
      Destination: 209.165.201.19 (SGW.protocol.DC.Local.1)
      PAYLOAD:
        Sx Session Establishment Response:
            Sx Session Establishment Response:
                Cause: 1
                OffendingIe: 0
                FSeid:
                    Seid: 10002
                    IPv4Address: 209.165.201.20
                CreatedTEndpt:
                    CreatedTEndpt[0]:
                        EndpointId: 1
                        FTeid:
                            Teid: 25270
                            IPv4Address: 209.165.201.1
                            ChooseId: 0
                    CreatedTEndpt[1]:
                        EndpointId: 2
                        FTeid:
                            Teid: 25271
                            IPv4Address: 209.165.200.226
                            ChooseId: 0
                MetaData: From:209.165.201.20:8805->To:209.165.201.19:10665
                Supi:
                Seid: 1297038098512740679
                Seqno: 4252
                Version: 0
                MsgPriority: false
                MsgPriorityVal: 0
```

```
                    Cmnid: 0
                    Rseid: 0
                    IntfType: 0
                    HdrLen: 0
                    MsgLen: 0
                    LoadControlInfo:
                        SeqNum: 0
                        Metric: 0
                        Valid: false
                    OverloadControlInfo:
                        SeqNum: 0
                        Metric: 0
                        Ociflag: 0
                        Valid: false

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.377609
      Message: GtpEpDecodeRPCResponse
      Description:  2071
      Source:
      Destination:
      PAYLOAD:

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.379043
      Message: S5 S8 Create Session Request
      Description: S5 S8 Create Session Request Message
      Source: 209.165.201.19
      Destination: 209.165.201.18
      PAYLOAD:
        S5 S8 Create Session Request:
            S5 S8 Create Session Request:
                Version: 2
                Pflag: false
                TEIDflag: true
                MsgPriority: false
                MsgLength: 221
                TIED: 0
                Seq: 66683
                MsgTypeId: 32
                MsgPriorityValue: 0
                Peer_IPv4_Flag: false
                Peer_IPv6_Flag: false
                MetaData: From:209.165.201.19:15001->To:209.165.201.18:2123
                Seid: 0
                Rseid: 0
                Cmnid: 0
                MsgType:
                    Create_Session_Request:
                        IMSI: 123456789012348
                        Recovery:
                            Value: 0
                        APN: intershat
                        AMBR:  UL: 232323 kbps, DL: 232323 kbps
                        MEI: 123456786666660
                        MSISDN: 223310101010101
                        Indication:
                            DAF: false
                            DTF: false
                            HI: false
```

```
        DFI: false
        OI: false
        ISRSI: false
        ISRAI: false
        SGWCI: false
        SQCI: false
        UIMSI: false
        CFSI: false
        CRSI: false
        P: false
        PT: false
        SI: false
        MSV: false
        RetLoc: false
        PBIC: false
        SRNI: false
        S6AF: false
        S4AF: false
        MBMDT: false
        ISRAU: false
        CCRSI: false
        CPRAI: false
        ARRL: false
        PPOF: false
        PPON_PPEI: false
        PPSI: false
        CSFBI: false
        CLII: false
        CPSR: false
        NSI: false
        UASI: false
        DTCI: false
        BDWI: false
        PSCI: false
        PCRI: false
        AOSI: false
        AOPI: false
        ROAAI: false
        EPCOSI: false
        CPOPCI: false
        PMTSMI: false
        S11TF: false
        PNSI: false
        UNACCSI: false
        WPMSI: false
        5GSIWK: false
        EEVRSI: false
        LTEMUI: false
        LTEMPI: false
        ENBCRSI: false
        TSPCMI: false
        PGBK: false
        PCPSI: false
        PCP: false
        PCPU: false
        N26_5GS: false
        RI_5GCN: false
        RS_5GCN: false
    PAA:
        PDN_Type: 1
        IPv4: 209.165.201.30
        IPv6_Prefix: 0
    RAT_Type:
        Value: 6
```

```
                                Serving_Network:
                                    MCC: 123
                                    MNC: 456
                                ULI:
                                    UliTai:  Mcc: 123, Mnc: 456, TAC: 2346
                                    UliEcgi:
                                        Mcc: 123
                                        Mnc: 456
                                        Eci: 1234567
                                FQ_TEID:
                                    SgwCntrl:
                                        IFace: 6
                                        TEID: 1375732039
                                        IPv4: 209.165.201.19
                                Bearer_Context_List:
                                    NumBearerCtxt: 1
                                    PbBearerCxt:
                                        PbBearerCxt[0]:
                                            BearerCtxType: 0
                                            EBI: 5
                                            Fqteid:
                                                SgwData:
                                                    IFace: 0
                                                    TEID: 25270
                                                    IPv4: 209.165.201.1
                                            BearerQos:
                                                PCI: true
                                                PL: 12
                                                PVI: true
                                                QCI: 6
                                                UL_MBR: 0 kbps
                                                DL_MBR: 0 kbps
                                                UL_GBR: 0 kbps
                                                DL_GBR: 0 kbps
                                                Arp: 113
                                                QciType: 0
                                Charging_Characteristics:
                                    Value:
                                        Value[0]: 210
                                        Value[1]: 4
                                        Value[2]: 0
                                        Value[3]: 0
                                PDN_Type:
                                    Value: 1
                                UE_Time_Zone:
                                    Time_Zone: 16
                                    Daylight_Saving_Time: 1
                                APN_Restriction:
                                    Value: 0
                                Selection_Mode:
                                    Value: 0
                                EPCO:
                                    Len: 5
                                    Value:
                                        Value[0]: 128
                                        Value[1]: 0
                                        Value[2]: 26
                                        Value[3]: 1
                                        Value[4]: 5

-------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/06/08 04:37:51.487884
```

```
      Message: GtpEpDecodeRPCResponse
      Description:  2071
      Source:
      Destination:
      PAYLOAD:

-------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/06/08 04:37:51.487884
      Message: GtpEpDecodeRPCIPCResponse
      Description:  2071
      Source:
      Destination:
      PAYLOAD:

-------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.475859
      Message: GtpEpDecodeRPCResponse
      Description:  2071
      Source:
      Destination:
      PAYLOAD:

-------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.476252
      Message: S5 S8 Create Session Response
      Description: S5 S8 Create Session Response Message
      Source: 209.165.201.18
      Destination: 209.165.201.19
      PAYLOAD:
        S5 S8 Create Session Response:
            S5 S8 Create Session Response:
                Version: 2
                Pflag: false
                TEIDflag: true
                MsgPriority: false
                MsgLength: 107
                TIED: 1375732039
                Seq: 66683
                MsgTypeId: 33
                MsgPriorityValue: 0
                Peer_IPv4_Flag: false
                Peer_IPv6_Flag: false
                MetaData: From:209.165.201.18:0->To:209.165.201.19:0
                Seid: 0
                Rseid: 0
                Cmnid: 0
                MsgType:
                    Create_Session_Response:
                        Cause:
                            Cause_Value: 16
                            PCE: false
                            BCE: false
                            OrigInd: false
                        Recovery:
                            Value: 100
                        AMBR:  UL: 10 kbps, DL: 20 kbps
                        PAA:
                            PDN_Type: 1
```

```
                                    IPv4: 209.165.201.26
                                    IPv6_Prefix: 0
                                FQ_TEID:
                                    PgwCntrl:
                                        IFace: 7
                                        TEID: 13210
                                        IPv4: 209.165.201.18
                                Bearer_Context_List:
                                    NumBearerCtxt: 1
                                    PbBearerCxt:
                                        PbBearerCxt[0]:
                                            BearerCtxType: 0
                                            EBI: 5
                                            Cause:
                                                Cause_Value: 16
                                                PCE: false
                                                BCE: false
                                                OrigInd: false
                                            Fqteid:
                                                PgwData:
                                                    IFace: 5
                                                    TEID: 13211
                                                    IPv4: 209.165.201.18
                                            ChrgId:
                                                Value: 303174163
                                APN_Restriction:
                                    Value: 1

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.483195
      Message: Sx Session Modification Request
      Description: Sx Session Modification Request Message from SGWC to SGWU
      Source: 209.165.201.19 (SGW.protocol.DC.Local.2)
      Destination: 209.165.201.20 (SGW.udp-proxy.DC.Local.0)
      PAYLOAD:
        Sx Session Modification Request:
            Sx Session Modification Request:
                UpdatePdr:
                    UpdatePdr[0]:
                        PdrId: 1
                        OuterHdrRem: 0
                        Precedence: 0
                        Pdi:
                            SrcIf: ACCESS
                            UeIp:
                                Src: false
                                Dst: false
                                IPv4Addr: 209.165.201.30
                            TEndpointId: 0
                            Valid: false
                        Qfi: 0
                UpdateFar:
                    UpdateFar[0]:
                        FarId: 2
                        ApplyAction:
                            Drop: false
                            Frwd: true
                            Buff: false
                            Nocp: false
                            Dupl: false
                            Valid: true
                        UpdateFwdParams:
```

```
                DestIf: ACCESS
                RedirectInfo:
                    AddrType: 0
                    Valid: false
                OuterHdr:
                    OuterHdrDesc: 256
                    Teid: 13211
                    IPv4Address: 209.165.201.18
                    Port: 0
                    Valid: true
                TEndptId: 0
                OuterPktTos: 0
                InnerPktTos: 0
                TosOpt:
                    CopyInner: false
                    CopyOuter: false
                SendTos: 0
                PfcpSmFlags:
                    Drobu: false
                    Qaurr: false
                    Sndem: false
                    Valid: false
                Valid: true
                NextHopId: 0
            UpdateDuplParams:
                DestIf: ACCESS
                OuterHdr:
                    OuterHdrDesc: 0
                    Teid: 0
                    IPv4Address: 209.165.201.30
                    Port: 0
                    Valid: false
                InterceptInfo:
                    InterceptId: 0
                    ChargingId: 0
                    SmfLiNodeId:
                        IpDesc: 0
                        IPv4Address: 209.165.201.30
                        Valid: false
                    PduSessionId: 0
                    Valid: false
                Valid: false
            BarId: 0
UplaneInacTimer: 0
MetaData: From:209.165.201.19:10002->To:209.165.201.20:8805
Supi:
Seid: 1297038098512740679
Seqno: 4248
Version: 0
MsgPriority: false
MsgPriorityVal: 0
Cmnid: 0
Rseid: 10002
IntfType: 0
HdrLen: 0
MsgLen: 0
PfcpSmFlags:
    Drobu: false
    Qaurr: false
    Sndem: false
    Valid: false
UserIDInfo:
    Valid: false
XHeaderInfo:
```

```
                    RatType:
                    Valid: false
                CfPolicyId:
                    PolicyId: 0
                    Valid: false
                GyStatus:
                    Valid: false
                    Value: false
                ChargingDisabled:
                    Valid: false
                    Value: false
                QueryInterface:
                    Valid: false
                    OfflineUrr: false
                    OnlineUrr: false
                    RadiusUrr: false
                    BearerUrr: false
                    SessUrr: false

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.549171
      Message: Sx Session Modification Response
      Description: Sx Session Modification Response Message from SGWU to SGWC
      Source: 209.165.201.20 (SGW.udp-proxy.DC.Local.0)
      Destination: 209.165.201.19 (SGW.protocol.DC.Local.2)
      PAYLOAD:
        Sx Session Modification Response:
            Sx Session Modification Response:
                Cause: 1
                OffendingIe: 0
                LoadControlInfo:
                    SeqNum: 0
                    Metric: 0
                    Valid: false
                OverloadControlInfo:
                    SeqNum: 0
                    Metric: 0
                    Ociflag: 0
                    Valid: false
                MetaData: From:209.165.201.20:8805->To:209.165.201.19:10002
                Supi:
                Seid: 1297038098512740679
                Seqno: 4248
                Version: 1
                MsgPriority: false
                MsgPriorityVal: 0
                Cmnid: 0
                Rseid: 0
                IntfType: 0
                HdrLen: 17
                MsgLen: 0

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.262224
      Message: S11 Create Session Request
      Description: S11 Create Session Request Message
      Source: 209.165.201.20
      Destination: 209.165.201.19
      PAYLOAD:
        S11 Create Session Request:
```

```
S11 Create Session Request:
    Version: 2
    Pflag: false
    TEIDflag: true
    MsgPriority: false
    MsgLength: 223
    TIED: 0
    Seq: 5842
    MsgTypeId: 32
    MsgPriorityValue: 0
    Peer_IPv4_Flag: false
    Peer_IPv6_Flag: false
    MetaData: From:209.165.201.20:2123->To:209.165.201.19:2123
    Seid: 0
    Rseid: 0
    Cmnid: 0
    MsgType:
        Create_Session_Request:
            IMSI: 123456789012348
            Recovery:
                Value: 100
            APN: intershat
            AMBR:  UL: 232323 kbps, DL: 232323 kbps
            MEI: 123456786666660
            MSISDN: 223310101010101
            Indication:
                DAF: false
                DTF: false
                HI: false
                DFI: false
                OI: false
                ISRSI: false
                ISRAI: false
                SGWCI: false
                SQCI: false
                UIMSI: false
                CFSI: false
                CRSI: false
                P: false
                PT: false
                SI: false
                MSV: false
                RetLoc: false
                PBIC: false
                SRNI: false
                S6AF: false
                S4AF: false
                MBMDT: false
                ISRAU: false
                CCRSI: false
                CPRAI: false
                ARRL: false
                PPOF: false
                PPON_PPEI: false
                PPSI: false
                CSFBI: false
                CLII: false
                CPSR: false
                NSI: false
                UASI: false
                DTCI: false
                BDWI: false
                PSCI: false
                PCRI: false
```

```
    AOSI: false
    AOPI: false
    ROAAI: false
    EPCOSI: false
    CPOPCI: false
    PMTSMI: false
    S11TF: false
    PNSI: false
    UNACCSI: false
    WPMSI: false
    5GSIWK: false
    EEVRSI: false
    LTEMUI: false
    LTEMPI: false
    ENBCRSI: false
    TSPCMI: false
    PGBK: false
    PCPSI: false
    PCP: false
    PCPU: false
    N26_5GS: false
    RI_5GCN: false
    RS_5GCN: false
PAA:
    PDN_Type: 1
    IPv4: 209.165.201.30
    IPv6_Prefix: 0
RAT_Type:
    Value: 6
Serving_Network:
    MCC: 123
    MNC: 456
ULI:
    UliTai:  Mcc: 123, Mnc: 456, TAC: 2346
    UliEcgi:
        Mcc: 123
        Mnc: 456
        Eci: 1234567
FQ_TEID:
    MmeCntrl:
        IFace: 10
        TEID: 25269
        IPv4: 209.165.201.20
    PgwCntrl:
        IFace: 7
        TEID: 0
        IPv4: 209.165.201.18
Bearer_Context_List:
    NumBearerCtxt: 1
    PbBearerCxt:
        PbBearerCxt[0]:
            BearerCtxType: 0
            EBI: 5
            Fqteid:
            BearerQos:
                PCI: true
                PL: 12
                PVI: true
                QCI: 6
                UL_MBR: 0 kbps
                DL_MBR: 0 kbps
                UL_GBR: 0 kbps
                DL_GBR: 0 kbps
                Arp: 113
```

```
                                        QciType: 0
                                Charging_Characteristics:
                                    Value:
                                        Value[0]: 210
                                        Value[1]: 4
                                        Value[2]: 0
                                        Value[3]: 0
                                PDN_Type:
                                    Value: 1
                                UE_Time_Zone:
                                    Time_Zone: 16
                                    Daylight_Saving_Time: 1
                                APN_Restriction:
                                    Value: 0
                                Selection_Mode:
                                    Value: 0
                                EPCO:
                                    Len: 5
                                    Value:
                                        Value[0]: 128
                                        Value[1]: 0
                                        Value[2]: 26
                                        Value[3]: 1
                                        Value[4]: 5

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.568954
      Message: GtpEpDecodeRPCResponse
      Description:  258
      Source:
      Destination:
      PAYLOAD:

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.569939
      Message: S11 Create Session Response
      Description: S11 Create Session Response Message
      Source: 209.165.201.19
      Destination: 209.165.201.20
      PAYLOAD:
        S11 Create Session Response:
            S11 Create Session Response:
                Version: 2
                Pflag: false
                TEIDflag: true
                MsgPriority: false
                MsgLength: 120
                TIED: 25269
                Seq: 5842
                MsgTypeId: 33
                MsgPriorityValue: 0
                Peer_IPv4_Flag: false
                Peer_IPv6_Flag: false
                MetaData: From:209.165.201.19:2123->To:209.165.201.20:2123
                Seid: 0
                Rseid: 0
                Cmnid: 0
                MsgType:
                    Create_Session_Response:
                        Cause:
```

```
                                    Cause_Value: 16
                                    PCE: false
                                    BCE: false
                                    OrigInd: false
                                Recovery:
                                    Value: 0
                                AMBR:  UL: 10 kbps, DL: 20 kbps
                                PAA:
                                    PDN_Type: 1
                                    IPv4: 209.165.201.26
                                    IPv6_Prefix: 0
                                FQ_TEID:
                                    PgwCntrl:
                                        IFace: 7
                                        TEID: 13210
                                        IPv4: 209.165.201.18
                                    SgwCntrl:
                                        IFace: 11
                                        TEID: 301990215
                                        IPv4: 209.165.201.19
                                Bearer_Context_List:
                                    NumBearerCtxt: 1
                                    PbBearerCxt:
                                        PbBearerCxt[0]:
                                            BearerCtxType: 0
                                            EBI: 5
                                            Cause:
                                                Cause_Value: 16
                                                PCE: false
                                                BCE: false
                                                OrigInd: false
                                            Fqteid:
                                                PgwData:
                                                    IFace: 5
                                                    TEID: 13211
                                                    IPv4: 209.165.201.18
                                                SgwData:
                                                    IFace: 1
                                                    TEID: 25271
                                                    IPv4: 209.165.200.226
                                            ChrgId:
                                                Value: 303174163
                                APN_Restriction:
                                    Value: 1

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.570132
      Message: S11 Create Session Response
      Description: S11 Create Session Response Message
      Source: 209.165.201.19
      Destination: 209.165.201.20
      PAYLOAD:
        S11 Create Session Response:
            S11 Create Session Response:
                Version: 2
                Pflag: false
                TEIDflag: true
                MsgPriority: false
                MsgLength: 120
                TIED: 25269
                Seq: 5842
                MsgTypeId: 33
```

```
                    MsgPriorityValue: 0
                    Peer_IPv4_Flag: false
                    Peer_IPv6_Flag: false
                    MetaData: From:209.165.201.19:2123->To:209.165.201.20:2123
                    Seid: 0
                    Rseid: 0
                    Cmnid: 0
                    MsgType:
                        Create_Session_Response:
                            Cause:
                                Cause_Value: 16
                                PCE: false
                                BCE: false
                                OrigInd: false
                            Recovery:
                                Value: 0
                            AMBR:  UL: 10 kbps, DL: 20 kbps
                            PAA:
                                PDN_Type: 1
                                IPv4: 209.165.201.26
                                IPv6_Prefix: 0
                            FQ_TEID:
                                PgwCntrl:
                                    IFace: 7
                                    TEID: 13210
                                    IPv4: 209.165.201.18
                                SgwCntrl:
                                    IFace: 11
                                    TEID: 301990215
                                    IPv4: 209.165.201.19
                            Bearer_Context_List:
                                NumBearerCtxt: 1
                                PbBearerCxt:
                                    PbBearerCxt[0]:
                                        BearerCtxType: 0
                                        EBI: 5
                                        Cause:
                                            Cause_Value: 16
                                            PCE: false
                                            BCE: false
                                            OrigInd: false
                                        Fqteid:
                                            PgwData:
                                                IFace: 5
                                                TEID: 13211
                                                IPv4: 209.165.201.18
                                            SgwData:
                                                IFace: 1
                                                TEID: 25271
                                                IPv4: 209.165.200.226
                                        ChrgId:
                                            Value: 303174163
                            APN_Restriction:
                                Value: 1

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.652708
      Message: Sx Session Modification Request
      Description: Sx Session Modification Request Message from SGWC to SGWU
      Source: 209.165.201.19 (SGW.protocol.DC.Local.0)
      Destination: 209.165.201.20 (SGW.udp-proxy.DC.Local.0)
      PAYLOAD:
```

```
Sx Session Modification Request:
    Sx Session Modification Request:
        UpdatePdr:
            UpdatePdr[0]:
                PdrId: 2
                OuterHdrRem: 0
                Precedence: 0
                Pdi:
                    SrcIf: ACCESS
                    UeIp:
                        Src: false
                        Dst: false
                        IPv4Addr: 209.165.201.30
                    TEndpointId: 0
                    Valid: false
                Qfi: 0
        UpdateFar:
            UpdateFar[0]:
                FarId: 1
                ApplyAction:
                    Drop: false
                    Frwd: true
                    Buff: false
                    Nocp: false
                    Dupl: false
                    Valid: true
                UpdateFwdParams:
                    DestIf: ACCESS
                    RedirectInfo:
                        AddrType: 0
                        Valid: false
                    OuterHdr:
                        OuterHdrDesc: 256
                        Teid: 25272
                        IPv4Address: 209.165.201.20
                        Port: 0
                        Valid: true
                    TEndptId: 0
                    OuterPktTos: 0
                    InnerPktTos: 0
                    TosOpt:
                        CopyInner: false
                        CopyOuter: false
                    SendTos: 0
                    PfcpSmFlags:
                        Drobu: false
                        Qaurr: false
                        Sndem: false
                        Valid: false
                    Valid: true
                    NextHopId: 0
                UpdateDuplParams:
                    DestIf: ACCESS
                    OuterHdr:
                        OuterHdrDesc: 0
                        Teid: 0
                        IPv4Address: 209.165.201.30
                        Port: 0
                        Valid: false
                    InterceptInfo:
                        InterceptId: 0
                        ChargingId: 0
                        SmfLiNodeId:
                            IpDesc: 0
```

```
                                        IPv4Address: 209.165.201.30
                                        Valid: false
                                    PduSessionId: 0
                                    Valid: false
                                Valid: false
                            BarId: 0
                        UplaneInacTimer: 0
                        MetaData: From:209.165.201.19:13486->To:209.165.201.20:8805
                        Supi:
                        Seid: 1297038098512740679
                        Seqno: 4245
                        Version: 0
                        MsgPriority: false
                        MsgPriorityVal: 0
                        Cmnid: 0
                        Rseid: 10002
                        IntfType: 0
                        HdrLen: 0
                        MsgLen: 0
                        PfcpSmFlags:
                            Drobu: false
                            Qaurr: false
                            Sndem: false
                            Valid: false
                        UserIDInfo:
                            Valid: false
                        XHeaderInfo:
                            RatType:
                            Valid: false
                        CfPolicyId:
                            PolicyId: 0
                            Valid: false
                        GyStatus:
                            Valid: false
                            Value: false
                        ChargingDisabled:
                            Valid: false
                            Value: false
                        QueryInterface:
                            Valid: false
                            OfflineUrr: false
                            OnlineUrr: false
                            RadiusUrr: false
                            BearerUrr: false
                            SessUrr: false

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.713891
      Message: Sx Session Modification Response
      Description: Sx Session Modification Response Message from SGWU to SGWC
      Source: 209.165.201.20 (SGW.udp-proxy.DC.Local.0)
      Destination: 209.165.201.19 (SGW.protocol.DC.Local.0)
      PAYLOAD:
        Sx Session Modification Response:
            Sx Session Modification Response:
                Cause: 1
                OffendingIe: 0
                LoadControlInfo:
                    SeqNum: 0
                    Metric: 0
                    Valid: false
                OverloadControlInfo:
```

```
                        SeqNum: 0
                        Metric: 0
                        Ociflag: 0
                        Valid: false
                    MetaData: From:209.165.201.20:8805->To:209.165.201.19:13486
                    Supi:
                    Seid: 1297038098512740679
                    Seqno: 4245
                    Version: 1
                    MsgPriority: false
                    MsgPriorityVal: 0
                    Cmnid: 0
                    Rseid: 0
                    IntfType: 0
                    HdrLen: 17
                    MsgLen: 0

------------------------------------------------

      Subscriber Id: imsi-123456789012348
      Timestamp: 2021/08/15 12:39:25.640179
      Message: S11 Modify Bearer Request
      Description: S11 Modify Bearer Request Message
      Source: 209.165.201.20
      Destination: 209.165.201.19
      PAYLOAD:
        S11 Modify Bearer Request:
            S11 Modify Bearer Request:
                Version: 2
                Pflag: false
                TEIDflag: true
                MsgPriority: false
                MsgLength: 58
                TIED: 301990215
                Seq: 5843
                MsgTypeId: 34
                MsgPriorityValue: 0
                Peer_IPv4_Flag: false
                Peer_IPv6_Flag: false
                MetaData: From:209.165.201.20:2123->To:209.165.201.19:2123
                Seid: 0
                Rseid: 0
                Cmnid: 0
                MsgType:
                    Modify_Bearer_Request:
                        RAT_Type:
                            Value: 6
                        Indication:
                            DAF: false
                            DTF: false
                            HI: false
                            DFI: false
                            OI: false
                            ISRSI: false
                            ISRAI: false
                            SGWCI: false
                            SQCI: false
                            UIMSI: false
                            CFSI: false
                            CRSI: false
                            P: false
                            PT: false
                            SI: false
                            MSV: false
```

```
                RetLoc: false
                PBIC: false
                SRNI: false
                S6AF: false
                S4AF: false
                MBMDT: false
                ISRAU: false
                CCRSI: false
                CPRAI: false
                ARRL: false
                PPOF: false
                PPON_PPEI: false
                PPSI: false
                CSFBI: false
                CLII: false
                CPSR: false
                NSI: false
                UASI: false
                DTCI: false
                BDWI: false
                PSCI: false
                PCRI: false
                AOSI: false
                AOPI: false
                ROAAI: false
                EPCOSI: false
                CPOPCI: false
                PMTSMI: false
                S11TF: false
                PNSI: false
                UNACCSI: false
                WPMSI: false
                5GSIWK: false
                EEVRSI: false
                LTEMUI: false
                LTEMPI: false
                ENBCRSI: false
                TSPCMI: false
                PGBK: false
                PCPSI: false
                PCP: false
                PCPU: false
                N26_5GS: false
                RI_5GCN: false
                RS_5GCN: false
            FQ_TEID:
            DelayValue:
                Value: 0
            Recovery:
                Value: 100
            Bearer_Context_List:
                NumBearerCtxt: 1
                PbBearerCxt:
                    PbBearerCxt[0]:
                        BearerCtxType: 0
                        EBI: 5
                        Fqteid:
                            ENbData:
                                IFace: 0
                                TEID: 25272
                                IPv4: 209.165.201.20

-------------------------------------------------
```

```
        Subscriber Id: imsi-123456789012348
        Timestamp: 2021/08/15 12:39:25.729811
        Message: S11 Modify Bearer Response
        Description: S11 Modify Bearer Response Message
        Source: 209.165.201.19
        Destination: 209.165.201.20
        PAYLOAD:
          S11 Modify Bearer Response:
              S11 Modify Bearer Response:
                  Version: 2
                  Pflag: false
                  TEIDflag: true
                  MsgPriority: false
                  MsgLength: 42
                  TIED: 25269
                  Seq: 5843
                  MsgTypeId: 35
                  MsgPriorityValue: 0
                  Peer_IPv4_Flag: false
                  Peer_IPv6_Flag: false
                  MetaData: From:209.165.201.19:2123->To:209.165.201.20:2123
                  Seid: 0
                  Rseid: 0
                  Cmnid: 0
                  MsgType:
                      Modify_Bearer_Response:
                          Cause:
                              Cause_Value: 16
                              PCE: false
                              BCE: false
                              OrigInd: false
                          Bearer_Context_List:
                              NumBearerCtxt: 1
                              PbBearerCxt:
                                  PbBearerCxt[0]:
                                      BearerCtxType: 0
                                      EBI: 5
                                      Cause:
                                          Cause_Value: 16
                                          PCE: false
                                          BCE: false
                                          OrigInd: false
                                      Fqteid:
                                          SgwData:
                                              IFace: 1
                                              TEID: 25271
                                              IPv4: 209.165.200.226

------------------------------------------------

        Subscriber Id: imsi-123456789012348
        Timestamp: 2021/08/15 12:39:25.730371
        Message: S11 Modify Bearer Response
        Description: S11 Modify Bearer Response Message
        Source: 209.165.201.19
        Destination: 209.165.201.20
        PAYLOAD:
          S11 Modify Bearer Response:
              S11 Modify Bearer Response:
                  Version: 2
                  Pflag: false
                  TEIDflag: true
                  MsgPriority: false
                  MsgLength: 42
```

```
                TIED: 25269
                Seq: 5843
                MsgTypeId: 35
                MsgPriorityValue: 0
                Peer_IPv4_Flag: false
                Peer_IPv6_Flag: false
                MetaData: From:209.165.201.19:2123->To:209.165.201.20:2123
                Seid: 0
                Rseid: 0
                Cmnid: 0
                MsgType:
                    Modify_Bearer_Response:
                        Cause:
                            Cause_Value: 16
                            PCE: false
                            BCE: false
                            OrigInd: false
                        Bearer_Context_List:
                            NumBearerCtxt: 1
                            PbBearerCxt:
                                PbBearerCxt[0]:
                                    BearerCtxType: 0
                                    EBI: 5
                                    Cause:
                                        Cause_Value: 16
                                        PCE: false
                                        BCE: false
                                        OrigInd: false
                                    Fqteid:
                                        SgwData:
                                            IFace: 1
                                            TEID: 25271
                                            IPv4: 209.165.200.226

------------------------------------------------

command terminated with exit code 124
  % Total    % Received % Xferd  Average Speed   Time
Time     Time  Current
                                 Dload  Upload   Total
Spent    Left  Speed
100   222  100    35  100   187   4375  23375
--:--:-- --:--:-- --:--:-- 27750
Stop Response Disabled mon_sub as part of timeout for
Cmd: --header Content-type:application/json --request
POST --data
{"commandname":"mon_sub","parameters":{"supi":"imsi-*","duration":300,
"enableTxnLog":false,"enableInternalMsg":false,"action":"stop","namespace":
"sgw","nf-service":"none","grInstance":0}}
http://oam-pod:8879/commands
```

モニタープロトコルの設定

この機能を設定するには、次の設定を使用します。

```
exec
   monitor protocol interface interface
      capture-duration capture_duration
      pcap [ Yes | No ]
      gr-instance gr_instance
      end
```

注：

- **monitor protocol interface** *interface*：PCAP がキャプチャされるインターフェイスを指定します。例：sbi、pfcp、gtpu、gtpc、gtp、radius。
- **capture-duration** *capture_duration*：PCAP がキャプチャされる期間を秒単位で指定します。デフォルト値は 300 秒です。
- **pcap [Yes | No]** ：PCAP ファイルの生成を設定します。デフォルトでは、PCAP 機能は無効になっています。
- **gr-instance** *gr_instance*：cnSGW-C がサブスクライバをモニターする GR インスタンスを指定します。

（注） GTP エンドポイントの IP が S5e および S11 インターフェイスの IP と同じ場合、プロトコル出力に一貫性がなくなり、メッセージを受信した S5 インターフェイスに S11 が表示されます。以下にエンドポイントの設定例を示します。

```
instance instance-id 1 endpoint gtp replicas 3 interface s5e vip-ip 209.165.201.22
instance instance-id 1 endpoint gtp replicas 3 interface s11 vip-ip 209.165.201.22
```

設定例

次に、設定例を示します。

```
monitor protocol interface pfcp,gtpc capture-duration 100 pcap yes
```

出力の例

次に、出力例を示します。

```
monitor protocol interface pcfp,gtpc capture-duration 100 pcap yes
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   231  100   101  100   130   6733   8666 --:--:-- --:--:-- --:--:-- 15400
Command: --header Content-type:application/json --request POST --data
{"commandname":"mon_pro","parameters":{"interface":"pcfp, gtpc","duration"
:100,"action":"start","enable_pcap": true,"grInstance":0}} http://oam-pod:8879/commands
Result start mon_pro, fileName -> logs/monprologs/sessintfname_pcfp,
gtpc_at_2021-06-06T06:15:39.005414271.txt
Starting to tail the monpro messages from file: logs/monprologs/sessintfname_pcfp,
gtpc_at_2021-06-06T06:15:39.005414271.txt
Defaulting container name to oam-pod.
Use 'kubectl describe pod/oam-pod-0 -n cn' to see all of the containers in this pod.
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:00.373101 +0000 UTC CaptureLength:59
Length:59 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528002dcc0e4000401
```

```
155510a01029c0a01029b3a9a084b001919634863000d000008d0
020175004900010005

Packet Dump:
 -- FULL PACKET DATA (59 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 2d cc 0e 40 00 40 11  55 51 0a 01 02 9c 0a 01  |.-..@.@.UQ......|
00000020  02 9b 3a 9a 08 4b 00 19  19 63 48 63 00 0d 00 00  |..:..K...cHc....|
00000030  08 d0 02 01 75 00 49 00  01 00 05                 |....u.I....|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..45..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..25..] Version=4 IHL=5 TOS=40 Length=45
Id=52238 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=21841 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 2d cc 0e 40 00  40 11 55 51 0a 01 02 9c  |E(.-..@.@.UQ....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..17..] SrcPort=15002 DstPort=2123(gtp-control)
Length=25 Checksum=6499}
00000000  3a 9a 08 4b 00 19 19 63                          |:..K...c|
--- Layer 4 ---
Payload 17 byte(s)
00000000  48 63 00 0d 00 00 08 d0  02 01 75 00 49 00 01 00  |Hc........u.I...|
00000010  05                                                |.|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15000 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:00.373236 +0000 UTC CaptureLength:72
Length:72 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528003a9af
c4000401186570a01029c0a01029a3a98084b0026196f 4824001a0000074a
00014f00490001000557000900865100009 70a01029c

Packet Dump:
 -- FULL PACKET DATA (72 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 3a 9a fc 40 00 40 11  86 57 0a 01 02 9c 0a 01  |.:..@.@..W......|
00000020  02 9a 3a 98 08 4b 00 26  19 6f 48 24 00 1a 00 00  |..:..K.&.oH$....|
00000030  07 4a 00 01 4f 00 49 00  01 00 05 57 00 09 00 86  |.J..O.I....W....|
00000040  51 00 00 97 0a 01 02 9c                          |Q.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..58..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..38..] Version=4 IHL=5 TOS=40 Length=58
Id=39676 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=34391 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 3a 9a fc 40 00  40 11 86 57 0a 01 02 9c  |E(.:..@.@..W....|
00000010  0a 01 02 9a                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..30..] SrcPort=15000(hydap) DstPort=2123(gtp-control)
```

```
Length=38 Checksum=6511}
00000000  3a 98 08 4b 00 26 19 6f                           |:..K.&.o|
--- Layer 4 ---
Payload 30 byte(s)
00000000  48 24 00 1a 00 00 07 4a  00 01 4f 00 49 00 01 00  |H$.....J..O.I...|
00000010  05 57 00 09 00 86 51 00  00 97 0a 01 02 9c        |.W....Q.......|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15000 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:02.462216 +0000 UTC CaptureLength:72
Length:72 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528003a9cdc4000401184770a01029c0a
01029a3a98084b0026196f4824001a0000074a00014f00490001000557000900865100000970a01029c

Packet Dump:
 -- FULL PACKET DATA (72 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 3a 9c dc 40 00 40 11  84 77 0a 01 02 9c 0a 01  |.:..@.@..w......|
00000020  02 9a 3a 98 08 4b 00 26  19 6f 48 24 00 1a 00 00  |..:..K.&.oH$....|
00000030  07 4a 00 01 4f 00 49 00  01 00 05 57 00 09 00 86  |.J..O.I....W....|
00000040  51 00 00 97 0a 01 02 9c                          |Q.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..58..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..38..] Version=4 IHL=5 TOS=40 Length=58
Id=40156 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=33911 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 3a 9c dc 40 00  40 11 84 77 0a 01 02 9c  |E(.:..@.@..w....|
00000010  0a 01 02 9a                                      |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..30..] SrcPort=15000(hydap) DstPort=2123(gtp-control)

Length=38 Checksum=6511}
00000000  3a 98 08 4b 00 26 19 6f                           |:..K.&.o|
--- Layer 4 ---
Payload 30 byte(s)
00000000  48 24 00 1a 00 00 07 4a  00 01 4f 00 49 00 01 00  |H$.....J..O.I...|
00000010  05 57 00 09 00 86 51 00  00 97 0a 01 02 9c        |.W....Q.......|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15000 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:04.563024 +0000 UTC CaptureLength:72
Length:72 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528003a9e4740004011830c0a01029c0a01029a3a98084b
0026196f4824001a0000074a00014f00490001000557000900865100000970a01029c
```

```
Packet Dump:
 -- FULL PACKET DATA (72 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 3a 9e 47 40 00 40 11  83 0c 0a 01 02 9c 0a 01  |.:.G@.@.........|
00000020  02 9a 3a 98 08 4b 00 26  19 6f 48 24 00 1a 00 00  |..:..K.&.oH$....|
00000030  07 4a 00 01 4f 00 49 00  01 00 05 57 00 09 00 86  |.J..O.I....W....|
00000040  51 00 00 97 0a 01 02 9c                          |Q.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..58..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..38..] Version=4 IHL=5 TOS=40 Length=58
Id=40519 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=33548 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 3a 9e 47 40 00  40 11 83 0c 0a 01 02 9c  |E(.:.G@.@.......|
00000010  0a 01 02 9a                                      |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..30..] SrcPort=15000(hydap)
DstPort=2123(gtp-control) Length=38 Checksum=6511}
00000000  3a 98 08 4b 00 26 19 6f                          |:..K.&.o|
--- Layer 4 ---
Payload 30 byte(s)
00000000  48 24 00 1a 00 00 07 4a  00 01 4f 00 49 00 01 00  |H$.....J..O.I...|
00000010  05 57 00 09 00 86 51 00  00 97 0a 01 02 9c        |.W....Q.......|


----------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15000 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:06.662105 +0000 UTC CaptureLength:72
Length:72 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528003a9eb440004011829f0a01029c0a01029a3a
98084b0026196f4824001a0000074a00014f00490001000557000900865100000970a01029c

Packet Dump:
 -- FULL PACKET DATA (72 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 3a 9e b4 40 00 40 11  82 9f 0a 01 02 9c 0a 01  |.:..@.@.........|
00000020  02 9a 3a 98 08 4b 00 26  19 6f 48 24 00 1a 00 00  |..:..K.&.oH$....|
00000030  07 4a 00 01 4f 00 49 00  01 00 05 57 00 09 00 86  |.J..O.I....W....|
00000040  51 00 00 97 0a 01 02 9c                          |Q.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..58..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..38..] Version=4 IHL=5 TOS=40 Length=58
Id=40628 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=33439 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 3a 9e b4 40 00  40 11 82 9f 0a 01 02 9c  |E(.:..@.@.......|
00000010  0a 01 02 9a                                      |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..30..] SrcPort=15000(hydap)
DstPort=2123(gtp-control) Length=38 Checksum=6511}
00000000  3a 98 08 4b 00 26 19 6f                          |:..K.&.o|
--- Layer 4 ---
Payload 30 byte(s)
```

```
00000000  48 24 00 1a 00 00 07 4a  00 01 4f 00 49 00 01 00  |H$.....J..O.I...|
00000010  05 57 00 09 00 86 51 00  00 97 0a 01 02 9c        |.W....Q.......|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:07.385688 +0000 UTC CaptureLength:59
Length:59 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528002dd2dd400040114e820a0102
9c0a01029b3a9a084b001919634863000d00000 8d00201750049000010005

Packet Dump:
 -- FULL PACKET DATA (59 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 2d d2 dd 40 00 40 11  4e 82 0a 01 02 9c 0a 01  |.-..@.@.N.......|
00000020  02 9b 3a 9a 08 4b 00 19  19 63 48 63 00 0d 00 00  |..:..K...cHc....|
00000030  08 d0 02 01 75 00 49 00  01 00 05                 |....u.I....|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..45..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..25..] Version=4 IHL=5 TOS=40 Length=45
Id=53981 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=20098 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 2d d2 dd 40 00  40 11 4e 82 0a 01 02 9c  |E(.-..@.@.N.....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..17..] SrcPort=15002 DstPort=2123(gtp-control)
Length=25 Checksum=6499}
00000000  3a 9a 08 4b 00 19 19 63                          |:..K...c|
--- Layer 4 ---
Payload 17 byte(s)
00000000  48 63 00 0d 00 00 08 d0  02 01 75 00 49 00 01 00  |Hc........u.I...|
00000010  05                                                |.|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15001 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:13.078691 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908045280029a11540004011804f0a01029c0
a01029a3a99084b0015195e400100090101ab000300010000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 a1 15 40 00 40 11  80 4f 0a 01 02 9c 0a 01  |.)..@.@..O......|
00000020  02 9a 3a 99 08 4b 00 15  19 5e 40 01 00 09 01 01  |..:..K...^@.....|
00000030  ab 00 03 00 01 00 00                             |.......|
```

```
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=41237 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=32847 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 29 a1 15 40 00  40 11 80 4f 0a 01 02 9c  |E(.)..@.@..O....|
00000010  0a 01 02 9a                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..13..] SrcPort=15001 DstPort=2123(gtp-control)
Length=21 Checksum=6494}
00000000  3a 99 08 4b 00 15 19 5e                          |:..K...^|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 01 01 ab 00  03 00 01 00 00           |@............|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15000 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:13.084971 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c0959080045280029d5a7400040114bbc0a01029c0a01029b3
a98084b0015195f400100090000df000300010000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 d5 a7 40 00 40 11  4b bc 0a 01 02 9c 0a 01  |.)..@.@.K.......|
00000020  02 9b 3a 98 08 4b 00 15  19 5f 40 01 00 09 00 00  |..:..K..._@.....|
00000030  df 00 03 00 01 00 00                             |.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=54695 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=19388 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 29 d5 a7 40 00  40 11 4b bc 0a 01 02 9c  |E(.)..@.@.K.....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..13..] SrcPort=15000(hydap) DstPort=2123(gtp-control)

Length=21 Checksum=6495}
00000000  3a 98 08 4b 00 15 19 5f                          |:..K..._|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 00 00 df 00  03 00 01 00 00           |@............|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
```

```
from 209.165.200.229:15002 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:14.485021 +0000 UTC CaptureLength:59
Length:59 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528002dd6794000401l4ae60a01029c0a01029b3
a9a084b001919634863000d000008d002017500490001000 5

Packet Dump:
 -- FULL PACKET DATA (59 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 2d d6 79 40 00 40 11  4a e6 0a 01 02 9c 0a 01  |.-.y@.@.J.......|
00000020  02 9b 3a 9a 08 4b 00 19  19 63 48 63 00 0d 00 00  |..:..K...cHc....|
00000030  08 d0 02 01 75 00 49 00  01 00 05                 |....u.I....|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..45..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..25..] Version=4 IHL=5 TOS=40 Length=45
Id=54905 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=19174 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 2d d6 79 40 00  40 11 4a e6 0a 01 02 9c  |E(.-.y@.@.J.....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..17..] SrcPort=15002 DstPort=2123(gtp-control)
Length=25 Checksum=6499}
00000000  3a 9a 08 4b 00 19 19 63                          |:..K...c|
--- Layer 4 ---
Payload 17 byte(s)
00000000  48 63 00 0d 00 00 08 d0  02 01 75 00 49 00 01 00  |Hc........u.I...|
00000010  05                                                |.|


--------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:18.078521 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c0959080045280029a1bb400040117fa90a01029c0a01029a3
a9a084b0015195e400100090201c9000300010000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 a1 bb 40 00 40 11  7f a9 0a 01 02 9c 0a 01  |.)..@.@.........|
00000020  02 9a 3a 9a 08 4b 00 15  19 5e 40 01 00 09 02 01  |..:..K...^@.....|
00000030  c9 00 03 00 01 00 00                              |.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=41403 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=32681 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 29 a1 bb 40 00  40 11 7f a9 0a 01 02 9c  |E(.)..@.@.......|
```

```
00000010  0a 01 02 9a                                        |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..13..] SrcPort=15002 DstPort=2123(gtp-control)
Length=21 Checksum=6494}
00000000  3a 9a 08 4b 00 15 19 5e                           |:..K...^|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 02 01 c9 00  03 00 01 00 00           |@............|


----------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15000 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:18.089434 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c0959080045280029d8cf4000401148940a0102
9c0a01029b3a98084b0015195f400100090000e0000300010000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 d8 cf 40 00 40 11  48 94 0a 01 02 9c 0a 01  |.)..@.@.H.......|
00000020  02 9b 3a 98 08 4b 00 15  19 5f 40 01 00 09 00 00  |..:..K..._@.....|
00000030  e0 00 03 00 01 00 00                              |.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=55503 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=18580 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 29 d8 cf 40 00  40 11 48 94 0a 01 02 9c  |E(.)..@.@.H.....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP    {Contents=[..8..] Payload=[..13..] SrcPort=15000(hydap) DstPort=2123(gtp-control)

Length=21 Checksum=6495}
00000000  3a 98 08 4b 00 15 19 5f                           |:..K..._|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 00 00 e0 00  03 00 01 00 00           |@............|


----------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:21.584999 +0000 UTC CaptureLength:59
Length:59 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528002ddb124000401146
4d0a01029c0a01029b3a9a084b001919634863000d00 0008d0020175004900010005
```

```
Packet Dump:
 -- FULL PACKET DATA (59 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 2d db 12 40 00 40 11  46 4d 0a 01 02 9c 0a 01  |.-..@.@.FM......|
00000020  02 9b 3a 9a 08 4b 00 19  19 63 48 63 00 0d 00 00  |..:..K...cHc....|
00000030  08 d0 02 01 75 00 49 00  01 00 05                 |....u.I....|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..45..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..25..] Version=4 IHL=5 TOS=40 Length=45
Id=56082 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=17997 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 2d db 12 40 00  40 11 46 4d 0a 01 02 9c  |E(.-..@.@.FM....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..17..] SrcPort=15002 DstPort=2123(gtp-control)
Length=25 Checksum=6499}
00000000  3a 9a 08 4b 00 19 19 63                          |:..K...c|
--- Layer 4 ---
Payload 17 byte(s)
00000000  48 63 00 0d 00 00 08 d0  02 01 75 00 49 00 01 00  |Hc........u.I...|
00000010  05                                                |.|


--------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15001 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:23.07887 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
000400010006005056c095908004528002 9a2c0400040117
ea40a01029c0a01029a3a99084b0015195e400100090101ac000300010000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 a2 c0 40 00 40 11  7e a4 0a 01 02 9c 0a 01  |.)..@.@.~.......|
00000020  02 9a 3a 99 08 4b 00 15  19 5e 40 01 00 09 01 01  |..:..K...^@.....|
00000030  ac 00 03 00 01 00 00                              |.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=41664 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=32420 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 29 a2 c0 40 00  40 11 7e a4 0a 01 02 9c  |E(.)..@.@.~.....|
00000010  0a 01 02 9a                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..13..] SrcPort=15001 DstPort=2123(gtp-control)
Length=21 Checksum=6494}
00000000  3a 99 08 4b 00 15 19 5e                          |:..K...^|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 01 01 ac 00  03 00 01 00 00           |@............|
```

```
------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:23.086144 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528002 9dbe740004011457c0a01029c0a01029b3
a9a084b0015195f400100090201760003000 10000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 db e7 40 00 40 11  45 7c 0a 01 02 9c 0a 01  |.)..@.@.E|......|
00000020  02 9b 3a 9a 08 4b 00 15  19 5f 40 01 00 09 02 01  |..:..K..._@.....|
00000030  76 00 03 00 01 00 00                              |v.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=56295 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=17788 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 29 db e7 40 00  40 11 45 7c 0a 01 02 9c  |E(.)..@.@.E|....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..13..] SrcPort=15002 DstPort=2123(gtp-control)
Length=21 Checksum=6495}
00000000  3a 9a 08 4b 00 15 19 5f                           |:..K..._|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 02 01 76 00  03 00 01 00 00           |@.....v......|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.227:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:28.079013 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c095908004528002 9a336400040117e2e0a01029c0
a01029a3a9a084b0015195e400100090201ca000300010000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 a3 36 40 00 40 11  7e 2e 0a 01 02 9c 0a 01  |.).6@.@.~.......|
00000020  02 9a 3a 9a 08 4b 00 15  19 5e 40 01 00 09 02 01  |..:..K...^@.....|
00000030  ca 00 03 00 01 00 00                              |.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
```

```
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=41782 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=32302 SrcIP=209.165.200.229 DstIP=209.165.200.227 Options=[] Padding=[]}
00000000  45 28 00 29 a3 36 40 00  40 11 7e 2e 0a 01 02 9c  |E(.).6@.@.~.....|
00000010  0a 01 02 9a                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..13..] SrcPort=15002 DstPort=2123(gtp-control)
Length=21 Checksum=6494}
00000000  3a 9a 08 4b 00 15 19 5e                           |:..K...^|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 02 01 ca 00  03 00 01 00 00           |@............|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:28.086562 +0000 UTC CaptureLength:55
Length:55 InterfaceIndex:0 AncillaryData:[]}

Packet Raw Bytes:
0004000100060050569c0959080045280029dfb14000401141b20a01029
c0a01029b3a9a084b0015195f400100090201770003000010000

Packet Dump:
 -- FULL PACKET DATA (55 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 29 df b1 40 00 40 11  41 b2 0a 01 02 9c 0a 01  |.)..@.@.A.......|
00000020  02 9b 3a 9a 08 4b 00 15  19 5f 40 01 00 09 02 01  |..:..K..._@.....|
00000030  77 00 03 00 01 00 00                              |w.......|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..41..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..21..] Version=4 IHL=5 TOS=40 Length=41
Id=57265 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=16818 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 29 df b1 40 00  40 11 41 b2 0a 01 02 9c  |E(.)..@.@.A.....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..13..] SrcPort=15002 DstPort=2123(gtp-control)
Length=21 Checksum=6495}
00000000  3a 9a 08 4b 00 15 19 5f                           |:..K..._|
--- Layer 4 ---
Payload 13 byte(s)
00000000  40 01 00 09 02 01 77 00  03 00 01 00 00           |@.....w......|


------------------------------------------------------------------------
InterfaceName = gtpc |  InterfaceIP = 209.165.200.229 | Filter = (tcp or udp or sctp)
and (port 2123 or (host 209.165.200.227 and port 2123)
or (host 209.165.201.11 and port 2123) or (host 209.165.200.228 and port 2123))
<<<<OUTBOUND
from 209.165.200.229:15002 to 209.165.200.228:2123
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2021-06-06 06:16:28.685112 +0000 UTC CaptureLength:59
Length:59 InterfaceIndex:0 AncillaryData:[]}
```

```
Packet Raw Bytes:
0004000100060050569c095908004528002de03240004011412d0a01029c0a01029b3a9a
084b001919634863000d000008d0020175004900010005

Packet Dump:
 -- FULL PACKET DATA (59 bytes) ------------------------------------
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00 45 28  |.......PV..Y..E(|
00000010  00 2d e0 32 40 00 40 11  41 2d 0a 01 02 9c 0a 01  |.-.2@.@.A-......|
00000020  02 9b 3a 9a 08 4b 00 19  19 63 48 63 00 0d 00 00  |..:..K...cHc....|
00000030  08 d0 02 01 75 00 49 00  01 00 05                 |....u.I....|
--- Layer 1 ---
Ethernet        {Contents=[..14..] Payload=[..45..] SrcMAC=00:50:56:9c:09:59
DstMAC=00:04:00:01:00:06 EthernetType=IPv4 Length=0}
00000000  00 04 00 01 00 06 00 50  56 9c 09 59 08 00        |.......PV..Y..|
--- Layer 2 ---
IPv4    {Contents=[..20..] Payload=[..25..] Version=4 IHL=5 TOS=40 Length=45
Id=57394 Flags=DF FragOffset=0 TTL=64 Protocol=UDP
Checksum=16685 SrcIP=209.165.200.229 DstIP=209.165.200.228 Options=[] Padding=[]}
00000000  45 28 00 2d e0 32 40 00  40 11 41 2d 0a 01 02 9c  |E(.-.2@.@.A-....|
00000010  0a 01 02 9b                                       |....|
--- Layer 3 ---
UDP     {Contents=[..8..] Payload=[..17..] SrcPort=15002 DstPort=2123(gtp-control)
Length=25 Checksum=6499}
00000000  3a 9a 08 4b 00 19 19 63                          |:..K...c|
--- Layer 4 ---
Payload 17 byte(s)
00000000  48 63 00 0d 00 00 08 d0  02 01 75 00 49 00 01 00  |Hc........u.I...|
00000010  05                                                |.|


------------------------------------------------------------------------
command terminated with exit code 124
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   172  100    43  100   129   8600  25800 --:--:-- --:--:-- --:--:-- 34400
Stop Response Already disabled mon_pro as part of timeout for Cmd: --header
Content-type:application/json --request POST --data
{"commandname":"mon_pro","parameters": {"interface":"pcfp,gtpc","duration":100, "action":

"stop","enable_pcap":true,"grInstance":0}} http://oam-pod:8879/commands
```

トランザクションメッセージの設定

トランザクションログを設定するには、次の設定を使用します。

```
config
   logging transaction message [ disable | enable ]
   end
```

注：

- **logging transaction message [disable | enable]**：トランザクションロギングのメッセージを設定します。enable に設定すると、トランザクションログと内部ログが結合されます。デフォルトでは、ログは disable になっています。

設定例

次に、設定例を示します。

```
logging transaction message enable
```

出力の例

次に、出力例を示します。

```
/opt/workspace/monsublogs# cat
'sgw.imsi-*_WithTxnLogs_TS_2020-12-22T15:16:58.158964842.txt'
Transaction Log received from Instance: smf.nodemgr.cn.cn.0
************* TRANSACTION: 00088 *************
TRANSACTION SUCCESS:
    Txn Type             : NmgrRersourceMgmtRequest(1025)
    Priority             : 1
    Session Namespace    : sgw(2)
LOG MESSAGES:
    2020/12/22 15:20:07.093 [TRACE] [infra.message_log.core] >>>>>>>>
IPC message
Name: NmgrRersourceMgmtRequest
MessageType: NmgrRersourceMgmtRequest
Key:
--body--
{"supi":"imsi-123456789012348","idReqType":1,"serviceUsertype":2}
    2020/12/22 15:20:07.093 [DEBUG] [nodemgr0.app.Int]
                GetSessionNamespace for txn id: 88, Type: 1025
    2020/12/22 15:20:07.093 [DEBUG] [nodemgr0.app.Int]
                GetSessionNamespace returned namespace : 2
    2020/12/22 15:20:07.093 [INFO] [nodemgr0.app.Int]
                RECEIVED REQUEST <-- NmgrRersourceMgmtRequest
    2020/12/22 15:20:07.093 [TRACE] [infra.message_log.core] <<<<<<<<

*********************************************
Transaction Log received from Instance: smf.protocol.cn.cn.0
************* TRANSACTION: 00050 *************
TRANSACTION SUCCESS:
    Txn Type             : 524(524)
    Priority             : 1
    Session Namespace    : sgw(2)

INCOMING REQUEST:
      Message: Sx Session Establishment Request
      Description: Sx Session Establishment Request Message from SGWC to SGWU
      Source: 209.165.200.229
      Destination: 209.165.200.228
      PAYLOAD:
        Sx Session Establishment Request:
            Sx Session Establishment Request:
                CreatePdr:
                    CreatePdr[0]:
                        PdrId: 1
                        Precedence: 0
                        Pdi:
                            SrcIf: CORE
                            UeIp:
                                Src: false
                                Dst: false
                                IPv4Addr: 209.165.201.30
                            TEndpointId: 1
                            Valid: true
                        OuterHdrRem: 0
                        FarId:
                            FarId[0]: 1
                        Qfi: 0
                        OuterHdrRemValid: false
                    CreatePdr[1]:
```

```
            PdrId: 2
            Precedence: 0
            Pdi:
                SrcIf: ACCESS
                UeIp:
                    Src: false
                    Dst: false
                    IPv4Addr: 209.165.201.30
                TEndpointId: 2
                Valid: true
            OuterHdrRem: 0
            FarId:
                FarId[0]: 2
            Qfi: 0
            OuterHdrRemValid: false
    CreateFar:
        CreateFar[0]:
            FarId: 1
            ApplyAction:
                Drop: true
                Frwd: false
                Buff: false
                Nocp: false
                Dupl: false
                Valid: true
            FwdParams:
                DestIf: ACCESS
                RedirectInfo:
                    AddrType: 0
                    Valid: false
                OuterHdr:
                    OuterHdrDesc: 0
                    Teid: 0
                    IPv4Address: 209.165.201.30
                    Port: 0
                    Valid: false
                TEndptId: 2
                OuterPktTos: 255
                InnerPktTos: 255
                TosOpt:
                    CopyInner: false
                    CopyOuter: false
                SendTos: 0
                PfcpSmFlags:
                    Drobu: false
                    Qaurr: false
                    Sndem: false
                    Valid: false
                Valid: true
            DuplParams:
                DestIf: ACCESS
                OuterHdr:
                    OuterHdrDesc: 0
                    Teid: 0
                    IPv4Address: 209.165.201.30
                    Port: 0
                    Valid: false
                InterceptInfo:
                    InterceptId: 0
                    ChargingId: 0
                    SmfLiNodeId:
                        IpDesc: 0
                        IPv4Address: 209.165.201.30
                        Valid: false
```

```
                    PduSessionId: 0
                    Valid: false
                Valid: false
            BarId: 0
        CreateFar[1]:
            FarId: 2
            ApplyAction:
                Drop: true
                Frwd: false
                Buff: false
                Nocp: false
                Dupl: false
                Valid: true
            FwdParams:
                DestIf: CORE
                RedirectInfo:
                    AddrType: 0
                    Valid: false
                OuterHdr:
                    OuterHdrDesc: 0
                    Teid: 0
                    IPv4Address: 209.165.201.30
                    Port: 0
                    Valid: false
                TEndptId: 1
                OuterPktTos: 255
                InnerPktTos: 255
                TosOpt:
                    CopyInner: false
                    CopyOuter: false
                SendTos: 0
                PfcpSmFlags:
                    Drobu: false
                    Qaurr: false
                    Sndem: false
                    Valid: false
                Valid: true
            DuplParams:
                DestIf: ACCESS
                OuterHdr:
                    OuterHdrDesc: 0
                    Teid: 0
                    IPv4Address: 209.165.201.30
                    Port: 0
                    Valid: false
                InterceptInfo:
                    InterceptId: 0
                    ChargingId: 0
                    SmfLiNodeId:
                        IpDesc: 0
                        IPv4Address: 209.165.201.30
                        Valid: false
                    PduSessionId: 0
                    Valid: false
                Valid: false
            BarId: 0
    CreateTEndpt:
        CreateTEndpt[0]:
            EndpointId: 1
            FTeid:
                Teid: 0
                IPv4Address: 209.165.201.30
                ChooseId: 0
            BearerLvlInfo:
```

```
                            Valid: 1
                            Qci: 6
                    CreateTEndpt[1]:
                        EndpointId: 2
                        FTeid:
                            Teid: 0
                            IPv4Address: 209.165.201.30
                            ChooseId: 0
                        BearerLvlInfo:
                            Valid: 1
                            Qci: 6
                PdnType: 0
                UplaneInacTimer: 0
                MetaData: From:209.165.200.229:11000->To:209.165.200.228:8805 Seqno:37
                            Supi: Seid:1224979349111832634 Cmnid:0 Rseid:0 IntfType:0
                UserIDInfo:
                    Valid: false
                XHeaderInfo:
                    RatType:
                    Valid: false
                CfPolicyId:
                    PolicyId: 0
                    Valid: false
                ChargingDisabled:
                    Valid: false
                    Value: false
                ChargingParams:
                    Valid: 0
                    GyOfflineChargingEnabled: 0

OUTGOING RESPONSE:
      Message: Sx Session Establishment Response
      Description: Sx Session Establishment Response Message from SGWU to SGWC
      Source: 209.165.200.228
      Destination: 209.165.200.229
      PAYLOAD:
        Sx Session Establishment Response:
            Sx Session Establishment Response:
                Cause: 1
                OffendingIe: 0
                FSeid:
                    Seid: 10002
                    IPv4Address: 209.165.200.228
                CreatedTEndpt:
                    CreatedTEndpt[0]:
                        EndpointId: 1
                        FTeid:
                            Teid: 2086
                            IPv4Address: 209.165.201.1
                            ChooseId: 0
                    CreatedTEndpt[1]:
                        EndpointId: 2
                        FTeid:
                            Teid: 2087
                            IPv4Address: 209.165.200.226
                            ChooseId: 0
                MetaData: From:209.165.200.228:8805->To:209.165.200.229:11000 Seqno:37
                            Supi: Seid:1224979349111832634 Cmnid:0 Rseid:0 IntfType:0
                LoadControlInfo:
                    SeqNum: 0
                    Metric: 0
                    Valid: false
                OverloadControlInfo:
                    SeqNum: 0
```

```
                    Metric: 0
                    Ociflag: 0
                    Valid: false

LOG MESSAGES:
    2020/12/22 15:20:10.934 [TRACE] [infra.message_log.core] >>>>>>>>

    2020/12/22 15:20:10.934 [DEBUG] [proto_ep.app.Int]
                             GetSessionNamespace for txn id: 50, Type: 524
    2020/12/22 15:20:10.934 [DEBUG]
                             [proto_ep.app.Int] GetSessionNamespace returned namespace :
 2
    2020/12/22 15:20:10.934 [DEBUG]
                           [infra.ipc_action.core] BG IPC will be executed after process
 continue.
    2020/12/22 15:20:10.934 [DEBUG]
                             [infra.ipc_action.core] BG IPC is executing after process
continue.
    2020/12/22 15:20:10.967 [DEBUG]
                             [infra.transaction.core] Response received for corelation
ID app_protocol-0_37
    2020/12/22 15:20:10.967 [DEBUG]
                             [infra.transaction.core]
                         (1) All pending ipc action finshed, hence waking up transaction

    2020/12/22 15:20:10.967 [TRACE]
                             [infra.message_log.core] <<<<<<<<
```

ログへのアクセス

モニターサブスクライバログにアクセスするには、/opt/workspace/logs/monsulogsの oam-pod に移動します。

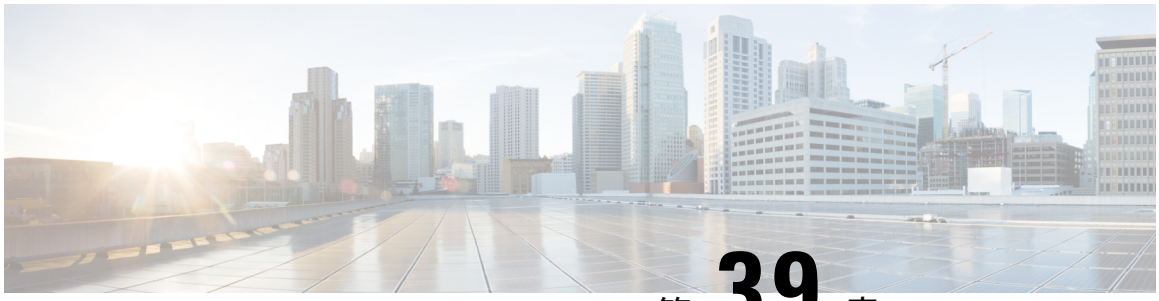
サンプルログ

```
root@oam-pod-0:/opt/workspace/logs/monsublogs# ls
none.imsi-123456789_TS_2021-06-05T06:19:12.682444275.txt
sgw.imei-352099001761480_TS_2021-06-05T13:07:41.774214146.txt.sorted
 none.imsi-123456789_TS_2021-06-05T06:20:39.751939118.txt
sgw.imei-352099001761480_TS_2021-06-05T13:48:51.868279985.txt
 none.imsi-123456789_TS_2021-06-06T06:22:16.015635407.txt
sgw.imei-352099001761480_TS_2021-06-05T13:48:51.868279985.txt.sorted
 sgw.imei-352099001761480_TS_2021-06-04T19:09:24.863985017.txt
sgw.imei-352099001761480_TS_2021-06-05T14:50:09.330635953.txt
 sgw.imei-352099001761480_TS_2021-06-04T19:09:24.863985017.txt.sorted
sgw.imei-352099001761480_TS_2021-06-05T14:50:09.330635953.txt.sorted
 sgw.imei-352099001761480_TS_2021-06-05T08:44:25.889632126.txt
sgw.imei-352099001761480_TS_2021-06-05T17:36:17.238331396.txt
 sgw.imei-352099001761480_TS_2021-06-05T08:44:25.889632126.txt.sorted
sgw.imei-352099001761480_TS_2021-06-05T17:36:17.238331396.txt.sorted
 sgw.imei-352099001761480_TS_2021-06-05T10:26:23.529652777.txt
'sgw.imsi-*_TS_2021-06-05T06:23:19.865508390.txt'
 sgw.imei-352099001761480_TS_2021-06-05T10:26:23.529652777.txt.sorted
'sgw.imsi-*_TS_2021-06-05T06:25:18.219875282.txt'
 sgw.imei-352099001761480_TS_2021-06-05T13:07:41.774214146.txt
```

モニタープロトコルログにアクセスするには、/opt/workspace/logs/monprologsの oam-pod に移動します。

サンプルログ

```
root@oam-pod-0:/opt/workspace/logs/monprologs# ls
sessintfname_gtpc_at_2021-06-05T06:28:18.310397784.pcap
sessintfname_pcfp,gtpc_at_2021-06-06T06:15:39.005414271.pcap
sessintfname_gtpc_at_2021-06-05T06:28:18.310397784.txt
sessintfname_pcfp,gtpc_at_2021-06-06T06:15:39.005414271.txt
```



第 39 章

複数 PDN の接続または接続解除手順

- 機能の概要と変更履歴 （481 ページ）
- 機能説明 （481 ページ）
- 機能の仕組み （482 ページ）

機能の概要と変更履歴

要約データ

表 *175* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *176* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2020.03.0

機能説明

cnSGW-C は、次の機能を処理します。

- UE による DN 接続の要求
- UE による PDN の切断の要求
- PGW が開始した PDN の切断
- 管理者が開始した切断

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

UE 要求 PDN 接続コールフロー

ここでは、UE 要求 PDN 接続コールフローについて説明します。

図 *94 : UE* 要求 *PDN* 接続コールフロー

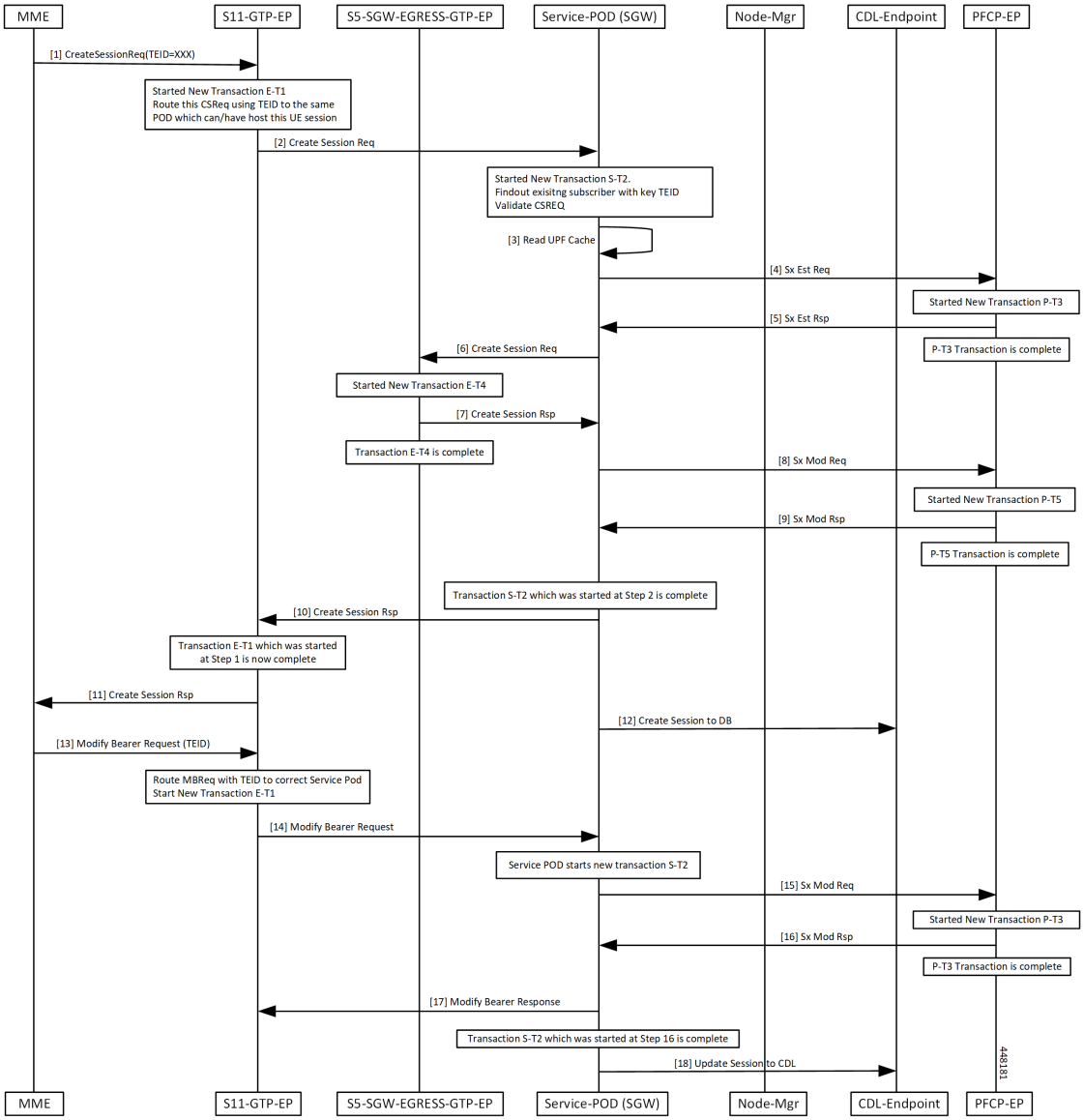


表 *177 : UE* 要求 *PDN* 接続コールフローの説明

ステップ	説明
1	MME が新しい PDN 接続の Create Session Request を、ゼロ以外の TEID を使用して S11-GTP-EP に送信します。

ステップ	説明
2	E-T1 トランザクションが開始しました。 S11-GTP-EP が受信した UDP メッセージを復号化し、メッセージを gGRPC メッセージに変換します。IMSI 値に基づいて、UE セッションを処理するそれぞれの SGW サービスポッドにメッセージが転送されます。 SGW サービスポッドが S11-GTP-EP から Create Session Request を受信します。
3	S-T2 トランザクションが開始しました。 SGW サービスポッドがローカル入力 TEID に基づいてサブスクライバコンテキストを検索し、Create Session Request のコンテンツを検証して PDN とサブスクライバ情報を更新します。 SGW サービスポッドが UPF キャッシュを読み取り、選択された UPF を提供します。
4	SGW サービスポッドが Sx Establishment Request を PFCP-EP に送信します。
5	P-T3 トランザクションが開始しました。 PFCP-EP が Sx Establishment Response を SGW サービスポッドに送信します。
6	P-T3 トランザクションが完了しました。 SGW サービスポッドが S11-U および S5-U TEID の詳細を使用して、Create Session Request を S5-SGW-EGRESS-GTP-EP に送信します。
7	E-T4 トランザクションが開始しました。 Create Session Response が検証され、S5-U リモート TEID が PDN で更新されます。 S5-SGW-EGRESS-GTP-EP が Create Session Response を SGW サービスポッドに送信します。
8	E-T4 トランザクションが完了しました。 SGW サービスポッドが Sx Modify Request を PFCP-EP に送信します。
9	P-T5 トランザクションが開始しました。 PFCP-EP が、タイマー T5 の期限切れ時に Sx Modify Response を SGW サービスポッドに送信します。
10	P-T5 および S-T2 トランザクションが完了しました。 GTP-EP が SGW サービスポッドから Create Session Response を受信します。
11	E-T1 トランザクションが完了しました。 MME が S11-GTP-EP から Create Session Response を受信します。

ステップ	説明
12	SGW サービスポッドが、CDL エンドポイント（データベース）で作成されたセッションを新しい PDN 情報で更新します。
13	MME が TEID を使用して Modify Bearer Request を S11-GTP-EP に送信します。
14	E-T1 トランザクションが開始しました。 S11-GTP-EP が Modify Bearer Request を SGW サービスポッドにルーティングします。 S11-GTP-EP が Modify Bearer Request を SGW サービスポッドに送信します。
15	S-T2 トランザクションが開始しました。 SGW サービスポッドが、タイマー S-T2 の期限切れ時に Sx Modify Request を PFCP-EP ポッドに送信します。
16	P-T3 トランザクションが開始しました。 PFCP-EP が、タイマー T3 の期限切れ時に Sx Modify Response を SGW サービスポッドに送信します。
17	R-T3 トランザクションが完了しました。 SGW サービスポッドが Modify Bearer Response を MME に送信します。
18	S-T2 トランザクションが完了しました。 SGW サービスポッドが更新セッションを CDL エンドポイントに送信します。

UE 要求または MME 要求の PDN 切断コールフロー

ここでは、UE または MME 要求の PDN 切断コールフローについて説明します。

図 *95 : UE* 要求または *MME* 要求の *PDN* 切断コールフロー

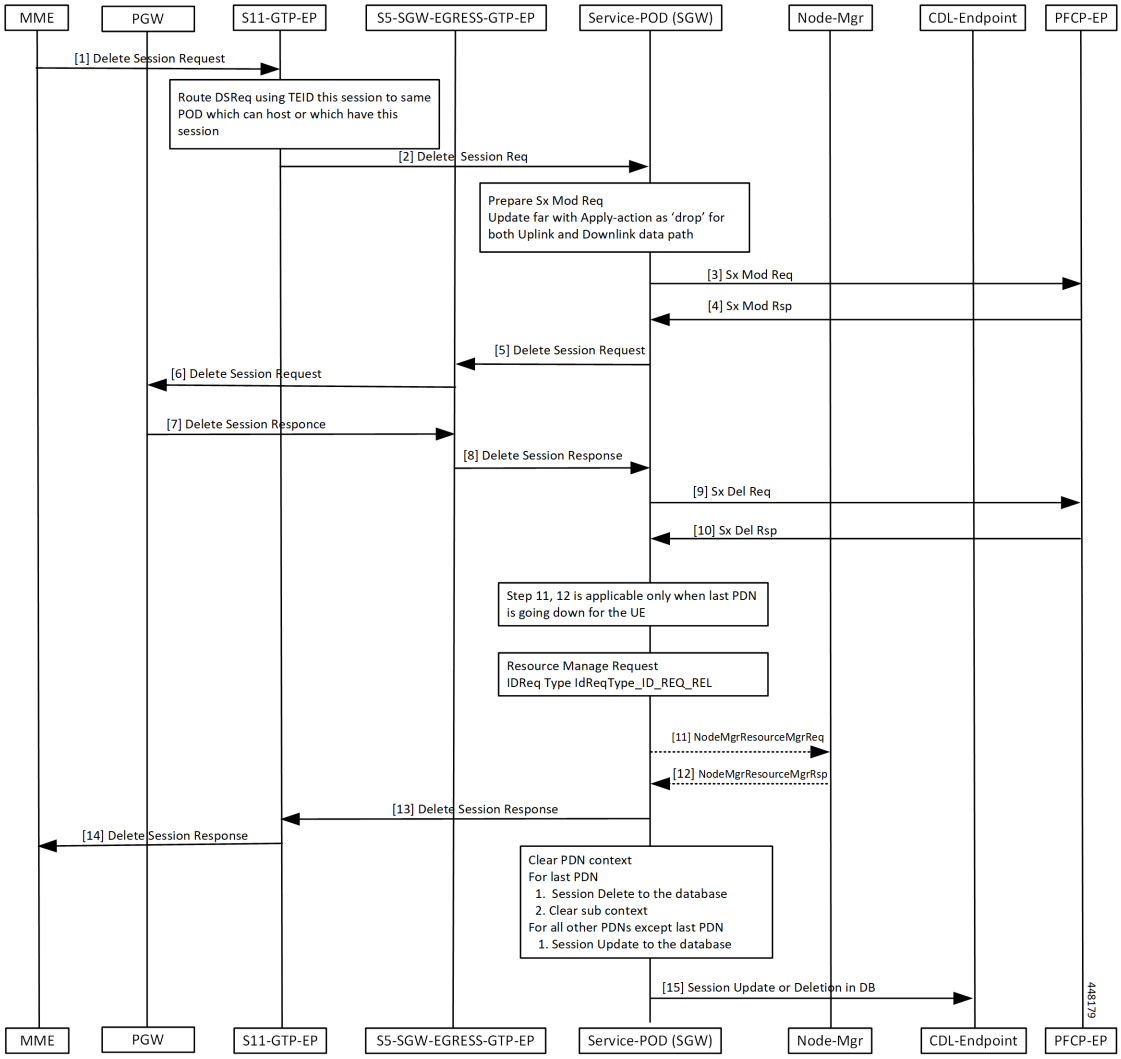


表 *178 : UE* 要求または *MME* 要求の *PDN* 切断コールフローの説明

ステップ	説明
1	MME が、他の PDN 切断について、セッション削除要求を S11-GTP-EP に送信します。
2	GTP-EP が受信した UDP メッセージを復号化し、メッセージを gRPC メッセージに変換します。 TEID 値に基づいて、gRPC メッセージが、UE セッションを処理できる SGW サービスポッドに転送されます。

ステップ	説明
3	SGW サービスポッドが、ローカル入力 TEID に従ってサブスクライバコンテキスト情報を検索します。 SGW サービスポッドが、セッション削除要求の内容を検証します。SGW サービスポッドが、適用アクションを DROP として PFCP-EP に Sx Modify Request を送信し、SGW-U でアップリンクまたはダウンリンクパケットをドロップします。
4	PFCP-EP が Sx Modify Response を SGW サービスポッドに送信します。
5	SGW サービスポッドは、セッション削除要求を S5-SGW-EGRESS-GTP-EP に転送します。
6	S5-SGW-EGRESS-GTP-EP が、UDP プロキシを介してセッション削除要求を PGW に転送します。
7	PGW が S5-SGW-EGRESS-GTP-EP にセッション削除応答を送信します。
8	S5-SGW-EGRESS-GTP-EP がセッション削除要求を SGW サービスポッドに転送します。
9	SGW サービスポッドがセッションの削除応答を検証し、Sx 削除要求を PFCP-EP に送信します。
10	SGW サービスポッドが、PFCP-EP から Sx セッション削除要求を受信します。
11	最後の PDN では、SGW サービスポッドが ID リリースの NodeMgrResourceManager 要求をノードマネージャに送信します。
12	ノードマネージャが ID をリリースし、確認応答を SGW サービスポッドに送信します。
13	SGW サービスポッドがセッション削除応答を S11-GTP-EP に送信します。
14	S11-GTP-EP がセッション削除応答を MME に転送します。
15	SGW サービスポッド • 最後の PDN 以外の PDN 情報を削除するために、データベース内のセッションを更新します。 • 最後の PDN のデータベースにセッション削除メッセージを送信します。

PGW 要求の切断コールフロー

ここでは、PGW 要求の切断コールフローについて説明します。

図 *96 : PGW* 要求の切断コールフロー

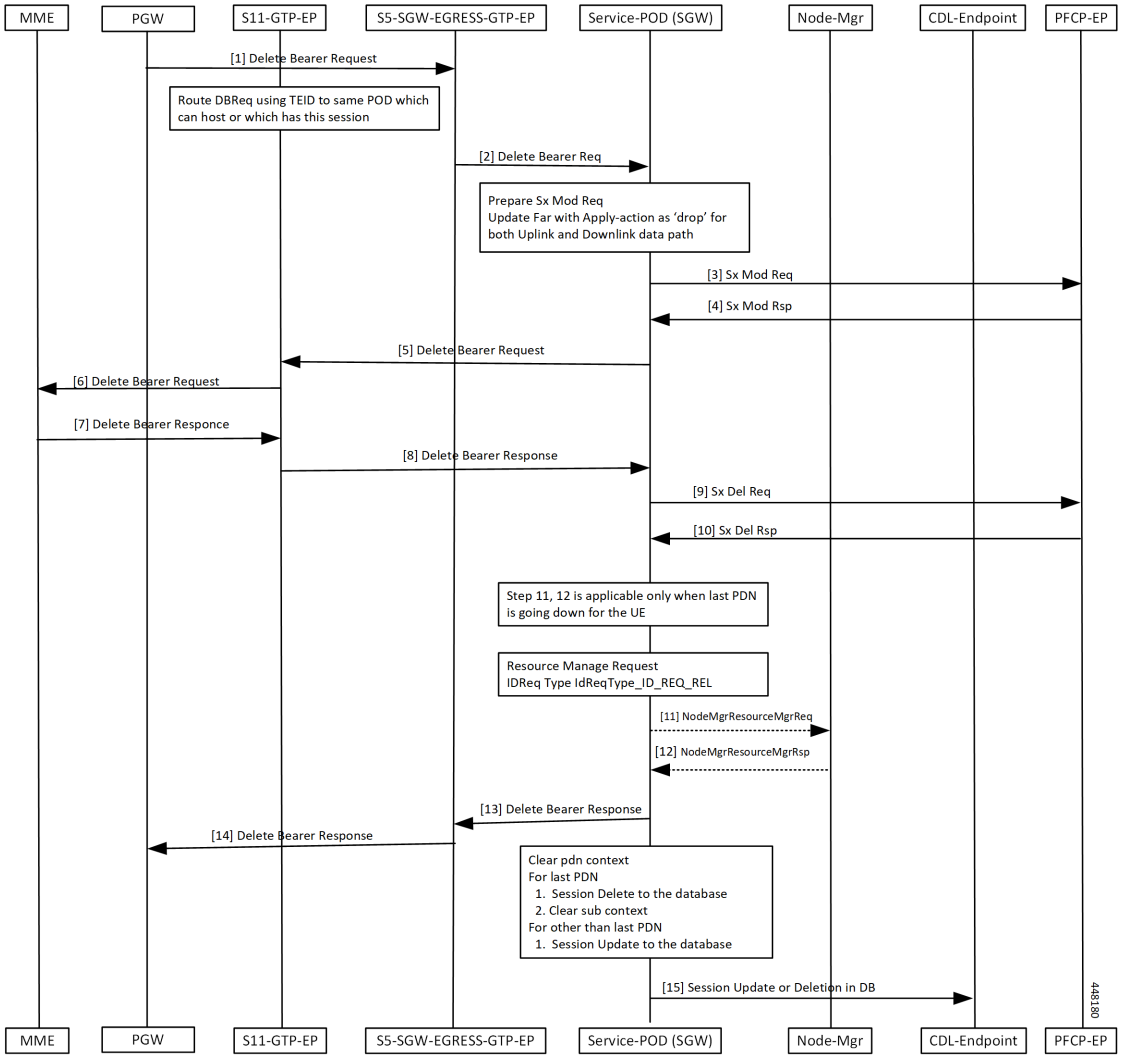


表 *179 : PGW* 要求の切断コールフローの説明

ステップ	説明
1	PGW が、他の PDN Disconnect に対して、Delete Bearer Request を S11-GTP-EP に送信します。
2	GTP-EP が受信した UDP メッセージを復号化し、メッセージを gRPC メッセージに変換します。 TEID 値に基づいて、UE セッションを処理できる SGW サービスポッドに gRPC メッセージが転送されます。

ステップ	説明
3	SGW サービスポッドが、ローカル入力 TEID に従ってサブスクライバコンテキスト情報を検索します。 SGW サービスポッドが Delete Bearer Request の内容を検証します。SGW サービスポッドが、適用アクションを DROP として Sx Modify Request を PFCP-EP に送信し、SGW-U でアップリンクまたはダウンリンクパケットをドロップします。
4	PFCP-EP が Sx Modify Response を SGW サービスポッドに送信します。
5	SGW サービスポッドが Delete Bearer Request を S5-SGW-EGRESS-GTP-EP に転送します。
6	S5-SGW-EGRESS-GTP-EP が UDP プロキシを介して Delete Bearer Request を PGW に転送します。
7	PGW が Delete Bearer Response を S5-SGW-EGRESS-GTP-EP に送信します。
8	S5-SGW-EGRESS-GTP-EP が Delete Bearer Response を SGW サービスポッドに転送します。
9	SGW サービスポッドが Delete Bearer Response を検証し、Sx Delete Request を PFCP-EP に送信します。
10	SGW サービスポッドが PFCP-EP から Sx Delete Bearer Response を受信します。
11	最後の PDN では、SGW サービスポッドが ID リリースの NodeMgrResourceManager 要求をノードマネージャに送信します。
12	ノードマネージャが ID をリリースし、確認応答を SGW サービスポッドに送信します。
13	SGW サービスポッドが Delete Bearer Response を S11-GTP-EP に送信します。
14	S11-GTP-EP が Delete Bearer Response を MME に転送します。
15	SGW サービスポッド • 最後の PDN 以外の PDN 情報を削除するために、データベース内のベアラーを更新します。 • 最後の PDN のデータベースに Delete Bearer メッセージを送信します。

第 40 章

パフォーマンス最適化のサポート

- 機能の概要と変更履歴 （492 ページ）
- 機能説明 （496 ページ）
- GTPC-EP から SGW サービスへの非同期 BG-IPC （497 ページ）
- バッチ ID の割り当て、リリース、および調整のサポート （497 ページ）
- キャッシュポッドの最適化 （500 ページ）
- CDL フラッシュ間隔およびセッション有効期限のチューニング設定 （501 ページ）
- DDN コールフロー最適化 （502 ページ）
- DDN タイムアウト設定 （507 ページ）
- Ops センターを使用したドメインベースのユーザー認可 （508 ページ）
- エッジエコーの実装 （511 ページ）
- ETCD ピア最適化サポート （513 ページ）
- 最適化された GTPv2 エンコーダおよびデコーダ （514 ページ）
- GR スプリットを使用した GTPC エンドポイント （516 ページ）
- S11 と S5 による GTPC エンドポイント インターフェイス分割 （517 ページ）
- GTPC IPC クロスラックサポート （520 ページ）
- サービス間ポッド通信 （530 ページ）
- MBR コールフロー最適化 （533 ページ）
- メンテナンス モード （544 ページ）
- アイドル/アクティブの部分的な CDL 更新コールフロー （546 ページ）
- DLDR スロットリングのサポートを含む PFCP セッションレポート （548 ページ）
- S11 インターフェイスでのセッション作成要求に対するスロットリングのサポート （552 ページ）
- 復元力処理 （555 ページ）
- ローミングピアパス管理の最適化 （559 ページ）
- DB データベース更新にフラグを設定する （568 ページ）
- プロトコルマイクロサービスに統合された UDP プロキシ機能 （569 ページ）

機能の概要と変更履歴

要約データ

表 *180 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当するプラットフォーム	SMI

機能のデフォルト設定	

	GTPC-EP から SGW サービスへの非同期 BG IPC：有効：常時オン バッチ ID の割り当て、リリース、調整のサポート：無効：有効にするには設定が必要です キャッシュポッドの最適化 CDL フラッシュ間隔およびセッション有効期限のチューニング設定：有効：無効にするには設定が必要です DDN コールフローの最適化：無効：有効にするには設定が必要です DDN タイムアウトの設定：無効：有効にするには設定が必要です エッジエコーの実装：有効：常時オン ETCD ピア最適化のサポート：有効：常時オン DB データベース更新へのフラグの設定：有効：常時オン GTPC インターフェイス分割：無効：有効にするには設定が必要です GTPC IPC クロスラックのサポート：無効：有効にするには設定が必要です サービス間ポッド通信：無効：有効にするには設定が必要です メンテナンスモード：無効：有効にするには設定が必要です MBR コールフローの最適化：無効：有効にするには設定が必要です 最適化された GTPv2 エンコーダおよびデコーダ：有効：常時オン アイドルアクティブ コールフローの部分的な CDL 更新：有効：常時オン DLDR スロットリングのサポートを含む PFCP セッションレポート：無効：有効にするには設定が必要です 復元力の処理：無効：有効にするには設定が必要です

	ローミングピアパス管理の最適化：無効：有効にするには設定が必要です UDP プロキシ機能のプロトコルマイクロサービスへのマージ：有効：無効にするには設定が必要です
関連資料	該当なし

マニュアルの変更履歴

表 *181* : マニュアルの変更履歴

改訂の詳細	リリース
次のサブ機能のサポートが導入されました。 • キャッシュポッドの最適化のサポートが追加されました。 • DLDR スロットリングのサポートを含む PFCP セッションレポート • 復元力処理	2023.01.0

改訂の詳細	リリース
次のサブ機能のサポートが導入されました。 • バッチIDの割り当て、リリース、および調整 • CDL フラッシュ間隔およびセッション有効期限のチューニング設定 • DDN コールフロー最適化 • DDN タイムアウト設定 • エッジエコーの実装 • ETCD ピア最適化サポート • DBデータベース更新にフラグを設定する • GR 分割 • GTPC インターフェイス分割 • GTPC IPC クロスラックサポート • サービス間ポッド通信 • IPv6 のサポートが導入されました。 • メンテナンス モード • ベアラー変更要求および応答（MBR）コールフローの最適化 • 追加の要求および応答メッセージ用に、最適化されたGTPv2エンコーダおよびデコーダが提供されています。 • UDP プロキシと GTPC-EP のマージ • UDP プロキシと PFCP-EP のマージ	2022.04.0
最初の導入。	2021.02.3

機能説明

この章では、パフォーマンス最適化機能について説明します。

一部のパフォーマンス最適化機能は、cnSGW-C と SMF に共通です。

SMF機能の詳細については、UCC 5G SMF設定および管理ガイド[英語]を参照してください。

GTPC-EP から SGW サービスへの非同期 BG-IPC

機能説明

cnSGW-C は、GTPC-EP ポッドのリソースの消費を抑えることで、GTPC-EP から cnSGW-C への非同期 BG-IPC コールをサポートします。

非同期 BG-IPC コールの場合、GTPC-EP ポッドは次を使用します。

- 要求メッセージを SGW サービスに送信するための、app-infra の SendRequestWithCallbackAsResponseWithRequestId API。
- SGW サービスから応答を受信するための GetIPCRequestCallbackResponseWithRequestId API。

バッチ ID の割り当て、リリース、および調整のサポート

機能説明

nodemgr は、接続状態にあるサブスクライバに一意の ID を割り当てます。サブスクライバが接続解除されると、一意の ID が nodemgr に解放されます。割り当て手順と割り当て解除手順が増えると、nodemgr のパフォーマンスが影響を受け、sgw-service はこれらの手順を完了するために長時間待機し続けます。

バッチ ID の割り当て、リリース、および調整サポート機能は、sgw-service と nodemgr 間のインタラクションを削減し、nodemgr のパフォーマンスを最適化するメカニズムを提供します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

バッチ IDの割り当て：

バッチ ID の割り当てには、次の手順が含まれます。

- sgw-service では、フリー ID をサブスクライバに割り当てることで ID ストアが管理されます。ID がストアで利用できない場合、sgw-service は Batch ID Allocation Request を nodemgr に送信します。
- nodemgr は応答で、ID Reserve Report Interval で 128 の ID を一括で返します。sgw-service は、nodemgr から受信した ID で ID ストアを更新し、ID Reserve Report Interval のタイマーを開始します。

- ID Reserve Report Interval で設定された期間の前にすべての ID が使用されている場合、sgw-service は、以前の要求からすべての ID を予約する通知とともに、ID Allocation Request を一括で nodemgr に送信します。
- sgw-service がすべての ID を割り当てる前に ID Reserve Report Interval タイマーが期限切れになった場合、sgw-service は Reserve Report Batch 操作を介して未使用の ID を nodemgr に返します。

バッチ ID のリリース：

バッチ ID のリリースには、次の手順が含まれます。

- sgw-service は、ID ストアが各 nodemgr に対して解放する ID を管理します。
- sgw-service は、ID の割り当てが解除されるたびに ID を ID ストアに返します。ID ストアがいっぱいの場合、sgw-service は Batch ID Release Request とリリースされた ID をそれぞれの nodemgr に送信します。
- sgw-service が ID ストアへの ID の追加を開始すると、ID リリースタイマーが開始します。
- バッチ ID がリリースされる前、またはバッチがいっぱいになる前に ID リリースタイマーが期限切れになった場合、sgw-service はリリースされた ID を nodemgr に送信します。

バッチ ID の調整

バッチ ID の調整は、サービスポッドと nodemgr ポッドの再起動時に発生します。

サービスポッドの再起動時：

1. サービスポッドがバッチ ID を受信し、ID を割り当てる前に応答しなくなった場合、nodemgr は Batch ID Reserve Request を取得できず、ID 予約手順がタイムアウトします。このようなシナリオでは、nodemgr は予約されていないか、割り当てられていない ID を CDL と調整します。サブスクライバに割り当てられていない ID は ID ストアに解放されます。
2. サービスポッドは、解放された ID を収集し、無応答になった場合 ID は nodemgr に解放します。このシナリオでは、ID はドロップされます。

nodemgr ポッドの再起動時：

1. 実行中の Batch ID Reserve Request と Batch ID Release Request に存在する ID はドロップされます。
2. nodemgr は、割り当てられた ID をバッチで cachemgr に通知します。ID を cachemgr に通知する前に nodemgr が応答しなくなった場合、nodemgr は再起動後に新しい ID の割り当てを開始します。nodemgr は、最後に割り当てられた ID とバッチサイズに基づいて ID を割り当てます。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
   sgw sgw_name
      resmgr-batch-operation [ disable | enable ]
      end
```

注：

resmgr-batch-operation [disable | enable]：バッチ操作を構成します。デフォルトでは、**resmgr-batch-operation** は無効です。

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計

バッチ ID の割り当ておよびリリースのサポート機能では、次の統計情報がサポートされています。

- sgw_resource_mgmt_stats：cnSGW-C リソース管理統計情報の合計数をキャプチャします。

  サンプル クエリ：

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",

id_req_type="id_batch_alloc",instance_id="0",service_name="sgw-service",status="attempted"}
 3
:sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",

id_req_type="id_batch_alloc",instance_id="0",service_name="sgw-service",status="success"}
 3
```

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1"

,id_req_type="id_batch_dealloc",instance_id="0",service_name="sgw-service",status="attempted"}
 2
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",id_req_type=

"id_batch_dealloc",instance_id="0",service_name="sgw-service",status="success"}
 2
```

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",

id_req_type="id_batch_dealloc_timeout",instance_id="0",service_name="sgw-service",status="attempted"}
 1
:sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",id_req_type

="id_batch_dealloc_timeout",instance_id="0",service_name="sgw-service",status="success"}
 1
```

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",

id_req_type="id_batch_release_timeout",instance_id="0",service_name="sgw-service",status="attempted"}
 1
-:sgw_resource_mgmt_stats{app_name="smf",cluster="Local",d-ata_center="DC",gr_instance_id="1",id_req_type

="id_batch_release_timeout",instance_id="0",service_name="sgw-service",status="success"}
 1
```

- nodemgr_rmgr_batch_reconcile_stats：調整のために送信されるバッチの総数をキャプチャします。

  サンプル クエリ：

```
nodemgr_rmgr_batch_reconcile_stats{app_name="smf",cluster="Local",data_center="DC",instance_id="0",
 service_name="nodemgr",status="success"} 1
```

- nodemgr_resource_mgmt_resp_stats：調整のために解放されたIDの合計数をキャプチャします。

  サンプル クエリ：

```
nodemgr_resource_mgmt_resp_stats{app_name="smf",cluster="Local",data_center="DC",error="",

gr_instance_id="0",instance_id="0",ip_ver_type="IP_TYPE_NONE",req_type="ID_REQ_REL_RECONCILE",
 service_name="nodemgr",status="success"} 16
```

バルク統計のサポートの詳細については、「UCC サービング ゲートウェイ コントロールプレーン機能メトリックリファレンス」を参照してください。

キャッシュポッドの最適化

機能説明

cnSGW-Cは、GTPCエンドポイントでのキャッシュポッドクエリを削減するためのキャッシュポッドの最適化をサポートしています。

アフィニティの取得クエリは、GTPCエンドポイントへの発信要求または応答メッセージでアフィニティ情報を受信するために使用されます。この最適化により、GTPCエンドポイントポッドは、今後の要求メッセージのキャッシュポッドにクエリを送信しません。

このアフィニティ情報を受信するために、GTPCエンドポイントへの発信要求または応答メッセージでアフィニティクエリが使用されます。この最適化により、GTPCエンドポイントポッドは、今後の要求メッセージのキャッシュポッドにクエリを送信しません。

以前のリリースでは、cnSGW-CがDDNを送信し、MMEからMBRを受信すると、GTPCエンドポイントはクエリをキャッシュポッドに送信してアフィニティ情報を取得する必要があり、

その後、cnSGW-C がアフィニティ情報を使用して、MBR を正しいサービスポッドに転送できていました。

この最適化により、余分なキャッシュポッドクエリを防ぐことができます。

CDL フラッシュ間隔およびセッション有効期限のチューニング設定

機能説明

デフォルトのサービスポッドパラメータを変更して、スループットパフォーマンスを微調整し、負荷パフォーマンスを最適化できます。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
   profile sgw sgw_name
      timers [ session-expiration-in-secs session_expiration |
affinity-expiration-in-secs affinity_expiration |
session-dbsync-interval-in-ms database_sync ]
      end
```

注：

- **session-expiration-in-secs** *session_expiration*：セッションがサービスポッドにキャッシュされる期間を指定します。*session_expiration* は、1 ～ 600 ミリ秒の範囲の値を受け入れます。デフォルト値は 30 ミリ秒です。
- **affinity-expiration-in-secs** *affinity_expiration*：サービスポッドやその他のポッドでセッションアフィニティキーが有効である期間を指定します。*affinity_expiration* は、1 ～ 1200 秒の範囲の値を受け入れます。デフォルトは 80 秒です。
- **session-dbsync-interval-in-ms** *database_sync* ：セッションがデータベースで同期されるまでの期間を指定します。*database_sync* は、1 ～ 10000 ミリ秒の値を受け入れます。デフォルトのタイムアウト値は 500 ミリ秒です。

設定例

次に、設定例を示します。

```
config
     profile sgw sgw1 [ timers session-expiration-in-secs 30 | affinity-expiration-in-secs
 80 | timers session-dbsync-interval-in-ms 500 ]
      end
```

DDN コールフロー最適化

機能説明

Downlink Data Notification（DDN）コールフロー最適化機能を使用すると、指定した期間、DDN手順の呼び出しを一時停止できます。この機能を使用すると、DDNを受信してUEがアクティブ状態に移行する前に、ネットワークが開始したサービスリクエスト手順がスケジュールされます。タイマーが期限切れになり、UE IDがアクティブであるか、またはModify Bearer Requestが進行中の場合、cnSGW は DDN 手順を無視します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C が、次のシナリオで DDN 手順を呼び出します。

- セッションがアイドル状態に急速に移行し、アップリンクとダウンリンクのデータが同時にトリガーされます。
- サブスクライバがアイドル状態のときにダウンリンクデータを受信すると、cnSGW-C が DDN を MME に送信します。UE をページングし、UE 状態をアクティブに変更するよう DDN が MME に通知します。
- サブスクライバがアイドル状態のときにアップリンクデータを受信すると、MMEがModify Bearer Request を SGW に送信し、UE 状態をアクティブに変更します。
- SGW サービスは、開始した DNN で Modify Bearer Request を受信した場合、DDN 手順を中止し、Modify Bearer Request を処理して UE 状態をアクティブに変更します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

現在の Downlink Data Notification 処理コールフロー

ここでは、現在の Downlink Data Notification 処理コールフローについて説明します。

図 *97* : 現在の *Downlink Data Notification* 処理コールフロー

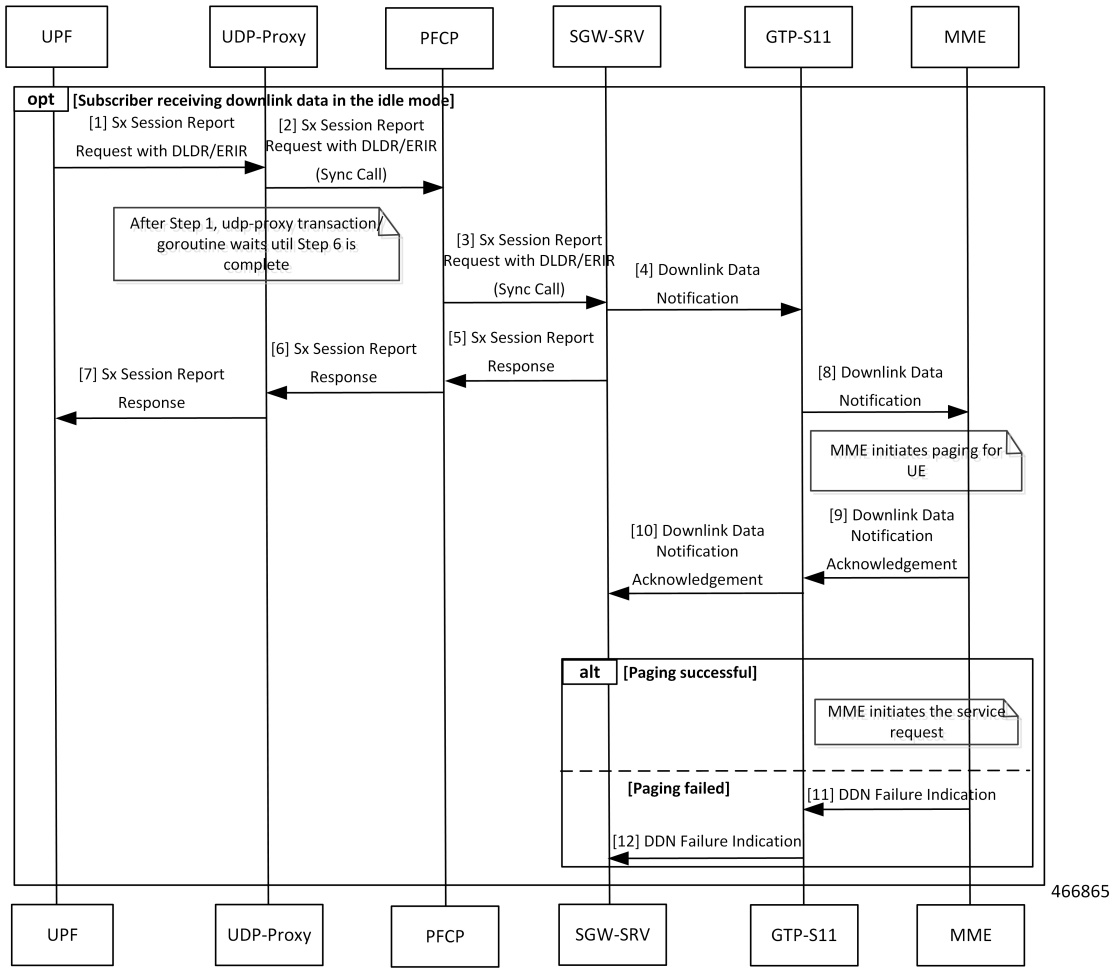


表 *182* : 現在の *Downlink Data Notification* 処理コールフローの説明

ステップ	説明
1	UPF が DLDR または ERIR を含む Sx Session Report Request を UDP プロキシに送信します。
2	UDP プロキシが DLDR または ERIR を含む Sx Session Report Request を PFCP に送信します。
3	PFCP が DLDR または ERIR を含む Sx Session Report Request を SGW-SRV に送信します。
4	SGW-SRV が Downlink Data Notification（DDN）を GTP-S11 に送信します。
5	SGW-SRV が Sx Session Report Response を PFCP に送信します。

ステップ	説明
6	PFCP が Sx Session Report Response を UDP プロキシに送信します。
7	UDP プロキシが Sx Session Report Response を UPF に送信します。
8	GTP-S11 が DDN 通知を MME に送信します。
9	MME が UE のページングを開始し、DDN 確認応答を GTP-S11 に送信します。
10	GTP-S11 が DDN 確認応答を SGW-SRV に送信します。
11	ページングが失敗すると、MME が DDN Failure Indication を GTP-S11 に送信します。
12	GTP-S11 が DDN Failure Indication を SGW-SRV に送信します。

内部 DDN 遅延タイマーによる DNN 処理のコールフロー

このセクションでは、内部 DDN 遅延タイマーによる DNN 処理のコールフローについて説明します。

図 *98* : 内部 *DDN* 遅延タイマーによる *DNN* 処理のコールフロー

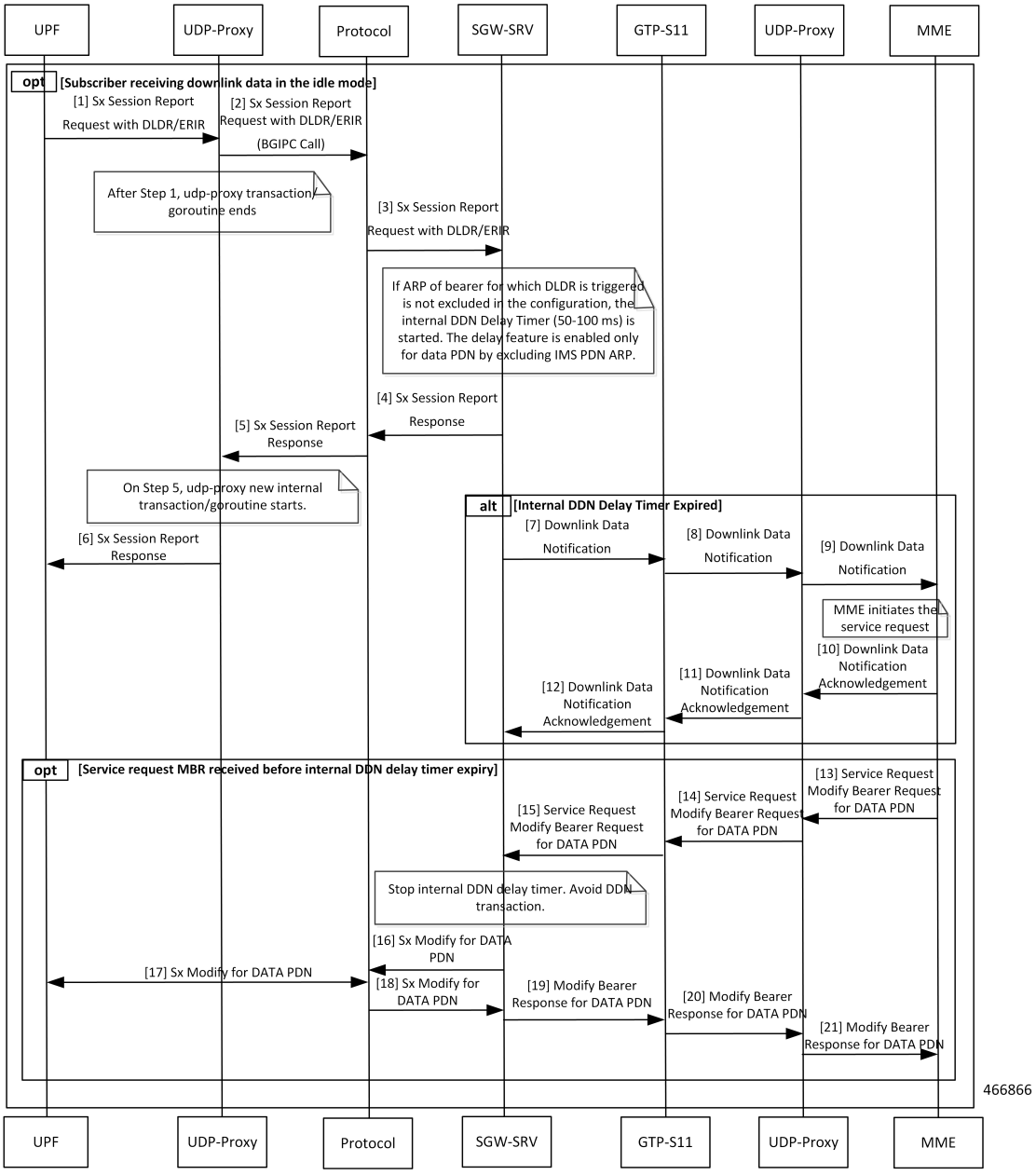


表 *183* : 内部 *DDN* 遅延タイマーによる *DNN* 処理のコールフローの説明

ステップ	説明
1	UPF が DLDR/ERIR を含む Sx Session Report Request を UDP-Proxy に送信します。
2	UDP プロキシが、DLDR/ERIR（BGIPC コール）を含む Sx セッションレポート要求をプロトコルに送信します。

ステップ	説明
3	プロトコルが、DLDR/ERIR を含む Sx セッションレポート要求を SGW-SRV に送信します。
4	SGW-SRV がプロトコルに Sx セッションレポート応答を送信します。
5	プロトコルが Sx セッション レポート要求を UDP プロキシに送信します。
6	UDP プロキシが、Sx セッションレポート要求を UPF に送信します。
7	SGW-SRV がダウンリンクデータ通知を GTP-S11 に送信します。
8	GTP-S11 がダウンリンクデータ通知を UDP プロキシに送信します。
9	UDP プロキシがダウンリンクデータ通知を MME に送信します。
10	MME が UE に対してページングを開始した後、MME はダウンリンクデータ通知の確認応答を UDP-Proxy に送信します。
11	UDP プロキシがダウンリンクデータ通知の確認応答を GTP-S11 に送信します。
12	GTP-S11 がダウンリンクデータ通知の確認応答を SGW-SRV に送信します。
13	MME がデータ PDN のベアラー変更要求を UDP プロキシに送信します。
14	UDP プロキシがデータ PDN のベアラー変更要求を GTP-S11 に送信します。
15	GTP-S11 がデータ PDN のベアラー変更要求を SGW-SRV に送信します。
16	SGW-SRV がデータ PDN のベアラー変更要求をプロトコルに送信します。
17	プロトコルと UPF が、データ PDN の Sx ベアラー変更要求を処理します。
18	プロトコルがデータ PDN の Sx 変更応答を SGW-SRV に送信します。
19	SGW-SRV がデータ PDN の Sx 変更応答を GTP-S11 に送信します。
20	GTP-S11 がデータ PDN の Sx 変更応答を UDP プロキシに送信します。
21	UDP プロキシがデータ PDN の Sx 変更応答を MME に送信します。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
   profile sgw sgw_name
      ddn { delay-exclude-arplist number_priorities | delay-timer
```

```
delay_duration }
       end
```

注：

- **delay-exclude-arplist** *number_priorities*：DDN の遅延から除外する必要がある割り当ての優先順位レベルおよび保持の優先順位（1 ー 15）を指定します。*number_priorities* は最大 8 つのエントリを受け入れることができます。
- **delay-timer** *delay_duration*：DDN 手順が遅延する期間を指定します。*delay_duration* はミリ秒単位（0 ～ 5000）の期間を受け入れます。デフォルトの期間は 0 で、これはタイマーが無効であることを示します。

設定例

次に、設定例を示します。

```
config
    profile sgw sgw1
       ddn delay-timer 100 delay-exclude-arplist [ 3 4 ]
       end
```

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計

DDN コールフロー最適化機能では、次の統計情報がサポートされています。

sgw_tmr_stats：停止、開始、および期限切れ状態の内部 DDN 遅延タイマー。

クエリ：

```
sgw_tmr_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",
instance_id="0",service_name="sgw-service",status="expired",timer_type="internal_ddn_delay"}
1

sgw_tmr_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",instance_id="0",
service_name="sgw-service",status="start",timer_type="internal_ddn_delay"} 2

sgw_tmr_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",instance_id="0",
service_name="sgw-service",status="stop",timer_type="internal_ddn_delay"} 1
```

DDN タイムアウト設定

機能説明

cnSGW-C では、cnSGW-C Ops センターを介して DDN タイムアウトとピアの応答なしの構成を指定できます。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
   profile sgw sgw_name
      ddn timeout-purge-session { true | false }
      end
```

注：

ddn timeout-purge-session { true | false }：MME が DDN 確認応答を送信しない場合にセッションを構成します。デフォルト値は false です。

Ops センターを使用したドメインベースのユーザー認可

機能説明

SMF と cnSGW-C は、Ops Center を使用したドメインベースのユーザー認証をサポートしています。ユーザー単位でアクセスを制御するには、Ops Center AAA で TACACS プロトコルを使用します。このプロトコルを使用すると、ルータまたは NAS へのアクセスを試みるユーザーを集中的に検証できます。

ルールリストで NETCONF アクセス制御（NACM）ルールを設定し、設定したルールを Ops Center の構成にマッピングして、グループを適切な運用許可にマッピングします。次の基準と製品に基づいた構成を使用します。

- NACM ルールと SMF ドメインベースのグループを使用して、SMF ベースの構成にアクセスのみ、または更新を許可するように Ops Center を設定します。
- NACM ルールと cSGW-C ドメインベースのグループを使用して、cSGW-C ベースの構成にアクセスのみ、または更新を許可するように Ops Center を設定します。
- NACM ルールと cSGW-C ドメインベースのグループを使用して、CCG ベースの構成にアクセスのみ、または更新を許可するように Ops Center を設定します。

（注） NSO サービスアカウントは、構成全体にアクセスできます。

機能の仕組み

Ops センターでこの機能設定をサポートするために、domain-based-services 設定が TACACS セキュリティ設定に追加されます。TACACS フローの変更は次のように機能します。

- **domain-based-services** パラメータを設定している場合、TACACS プロセスに送信される設定済みのユーザー名は、ユーザー ID をユーザー ID と ドメインに分割します。ドメイン

デリミタである分割文字は、domain-based-services で設定されます。これらの分割文字は「@」、「/」、または「\」であり、ドメインおよびユーザーID情報を取得するために次の形式で使用されます。

- @ — <user id>@<domain>
- / — <domain>/<user id>
- \ — <domain>\<user id>

- TACACSは、既存のフローに従って認証および認可します。ただし、domain-based-services機能が有効になっていて、TACACSがユーザーを認証および認可する場合、次のステップが TACACS フロー手順に追加されます。
  - Network Services Orchestrator（NSO）が NSO サービスアカウントとしてログインすると、そのセッションは、**domain-based-services nso-service-account group** *group-name* で設定した特定の NACM グループを受け取ります。これは、機能的にはNSOの動作と同じです。
  - 指定したドメインがグループマッピングに存在する場合は、**domain-based-services domain-service** *domain* **group** *group-name* で設定したNACMグループが適用されます。
  - ユーザーにドメインがない場合、またはドメインがドメインからグループへのマッピングに存在しない場合、**domain-based-services no-domain group** *group-name* で設定した **no-domain** NACM グループが適用されます。**no-domain** 設定が存在しない場合、ユーザー値は拒否されます。

この機能を有効にするには、次のオプションを指定して **domain-based-services** CLI コマンドを設定する必要があります。

- NSO サービスアカウント
- ドメインサービス
- ドメインデリミタ
- ドメインなし

機能設定

Opsセンターを使用してドメインベースのユーザー認証を有効にするには、次の設定例を使用します。

```
config
  tacacs-security domain-based-services [ domain-delimiter delimiter_option
 | domain-service domain_service_name [ group service_group_name  ] | no-domain
 group service_group_name | nso-service-account [ group service_group_name |
id service_account_id ] ]
   end
```

注：

- **domain-based-services [domain-delimiter** *delimiter_option* **| domain-service** *domain_service_name* **[group** *service_group_name* **] | no-domain group** *service_group_name* **| nso-service-account [group** *service_group_name* **| id** *service_account_id* **]]**：必要な domain-based-services 値を設定します。**domain-based-services** には次のオプションが含まれます。
  - **domain-delimiter**：ドメインの決定に使用するデリミタを指定します。このオプションは必須であり、次の値を使用できます。
    - @：domain-delimiter が「@」の場合、ユーザー値は <user> @<domain> 形式になります。
    - /：domain-delimiter が「/」の場合、ユーザー値は <domain>/<user> 形式になります。
    - \：domain-delimiter が「\」の場合、ユーザー値は <domain>\<user> 形式になります。
  - **domain-service**：ドメインおよびそのドメインのグループマッピングのリストを指定します。キーはドメインの名前で、グループはドメインに割り当てられたグループです。このリストの少なくとも 1 つのオプションを設定する必要があります。
  - **no-domain**：ドメインを持たないグループを指定します。またはドメインがドメインサービスマッピングで使用できない場合、このグループが受け入れ応答で送信されます。
  - **nso-service-account**：ID とグループを持つNSOサービスアカウントを指定します。このパラメータを設定する場合は、ID フィールドとグループフィールドを設定する必要があります。ID とグループには文字列値が必要です。

設定例

次に、TACACS セキュリティモードでのドメインベースのユーザー認証の例を示します。

```
config
   tacacs-security domain-based-services nso-service-account id nsid
      tacacs-security domain-based-services nso-service-account group nso-group
   tacacs-security domain-based-services no-domain group read-operational
   tacacs-security domain-based-services domain-delimiter @
   tacacs-security domain-based-services domain-service etcd
      group etcd
exit
tacacs-security domain-based-services domain-service sgw
   group sgw_1
exit
tacacs-security domain-based-services domain-service smf
   group smf
exit
```

設定の確認

構成を確認するには、次の show コマンドを使用します。

show running-config tacacs-security

この show コマンドの出力には、TACACS セキュリティ内のドメインベースのサービスのすべての構成が表示されます。

```
[smf] smf# show running-config tacacs-security
tacacs-security service smf
tacacs-security server 1
address 209.165.200.234
key $8$+twbdL2ZCgmjVswgp7kFJp8+SMXDjQRTZgoPVa3oEwY=
exit
tacacs-security domain-based-services nso-service-account id nsid
tacacs-security domain-based-services nso-service-account group nso-group
tacacs-security domain-based-services no-domain group read-operational
tacacs-security domain-based-services domain-delimiter @
tacacs-security domain-based-services domain-service etcd
group etcd
exit
tacacs-security domain-based-services domain-service sgw
group sgw_1
exit
tacacs-security domain-based-services domain-service smf
group smf
exit
```

エッジエコーの実装

機能説明

非マージモードでは、udp-proxy ポッドはエンドポイントとして機能し、gtpc-ep はピアノードからのエコー要求に応答します。

gtpc-ep は、システムが多数の入力CEPS を受信した際にトラフィックを処理し、その結果として、gtpc-ep が udp-proxy からメッセージを取得する速度と、udp-proxy がメッセージを受け取る速度との間に不整合が生じます。

gtpc-ep がロードされると、udp-proxy と gtpc-ep の間のキューがいっぱいになり、udp-proxy の一部のメッセージがドロップされることがあります。これらがエコー要求メッセージである場合、エコー応答を受信していないため、ピアはパス障害を検出します。さらに、ピアは sgw サービスに送信されたすべてのセッションをクリアします。

機能の仕組み

ここでは、この機能の仕組みを説明します。

nodemgr は次の手順でエコー要求を処理します。

- nodemgr には、各 GR インスタンス ID と GTPC ピアの自己再起動カウンタキャッシュが保持されます。
- UDP プロキシポッドがピアからエコー要求を受信し、自己再起動カウンタ値が自己再起動カウンタキャッシュで利用できない場合、UDP プロキシポッドがエコー要求を GTPC-EP に転送します。

- GTPC-EP が UDP プロキシ メッセージ メタデータの一部としてエコー応答で自己再起動カウンタを送信します。UDPプロキシが自己再起動カウンタを自己再起動カウンタキャッシュに保存します。UDPプロキシは、ピアからエコー要求を受信し、自己再起動カウンタ値が自己再起動カウンタキャッシュで利用可能な場合、再起動カウンタとともにエコー応答を送信します。
- UDPプロキシがエコー要求メッセージをGTPC-EPに転送します。GTPC-EPがエコー要求を処理し、必要に応じてnodemgrに転送します。
- ピア再起動カウンタ値が変更されると、nodemgrはパス障害を検出します。
- エコー応答では、GTPC-EP が UDP プロキシ メッセージ メタデータの自己再起動カウンタをUDPプロキシに送信します。自己再起動カウンタが自己再起動カウンタキャッシュに保存されているカウンタとは異なる場合、UDPプロキシはキャッシュ内の自己再起動カウンタを更新し、GTPC-EP から受信したエコー応答をドロップします。

（注） GTPC-EP がマージモードで開始された場合、エッジエコー機能はサポートされません。

Heartbeat

GTPV2インターフェイスのエコー要求メッセージとエコー応答メッセージを処理するために、GTPC-EP と UDP プロキシポッドの間にハートビートキューが実装されます。ハートビートキューは、PFCPインターフェイスのプロトコルとUDPプロキシポッド間のハートビート要求およびハートビート応答メッセージに関与します。

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートに関して説明します。

バルク統計サポート

エッジエコー実装機能では、次の統計情報がサポートされています。

- ハートビート キュー ステータス：

```
sum(irate(ipc_response_total{rpc_name~=".ipc_stream_hb."}[10s])) by
(service_name, instance_id, status, status_code, rpc_name, dest_host)
```

- EdgeEcho メッセージを確認します。

```
sum(irate(udp_proxy_msg_total{ message_name ="edge_echo"}[30s])) by
(message_name, message_direction, status)
```

ハートビートキューおよびEdgeEchoメッセージを有効にするには、以下を使用してudp_proxy_msg_totalのトレースレベル統計を構成します。

```
infra metrics verbose application
   metrics udp_proxy_msg_total level trace
   exit
```

（注） ハートビートおよびEdgeEcho統計を有効にすると、udp-proxyポッドのパフォーマンスが低下する可能性があります。

ETCD ピア最適化サポート

機能説明

多数の GTPC ピアが SMF または cnSGW-C に接続されている場合、ETCD のパフォーマンスに影響が出ます。各ピアは ETCD のレコードと見なされ、タイムスタンプはピア単位で 30 秒ごとに更新されるため、ETCD が継続的に更新され、システム全体のパフォーマンスに影響を与える大量のトラフィックが生成されます。

ETCD ピア最適化機能は、ピア管理の最適化を促進し、ETCD に対するパフォーマンスの影響の軽減を可能にします。

機能の仕組み

ここでは、この機能の仕組みを説明します。

各ピアをETCDレコードエントリと見なさず、各ピアのIPアドレスのハッシュ値に基づいて、複数のピアをピアグループとしてグループ化します。これにより、ETCDのエントリの数が減少します。デフォルトでは、最大200のピアグループを作成できます。ピアグループ内のピアに関連する変更が行われた場合：

- 新しいピアの場合、ピアグループは ETCD ですぐに保持されます。
- 既存のピアのタイムスタンプが変更された場合、ピアグループは3秒ごとに1回更新されます。この更新により、次の影響があります。
  - 各ピアグループ内でタイムスタンプが変更された多数のピアに対して、累積的なグループアップデートが行われます。
  - ETCD の頻繁な更新が減少します。

最適化された GTPv2 エンコーダおよびデコーダ

機能説明

cnSGW-C は、次に対して、最適化された GTPv2 エンコーダおよびデコーダを提供します。

- Modify Bearer Request メッセージを受信後にサブスクライバがアクティブ状態に移行している場合の Modify Bearer Request および Response メッセージ。
- Release Access Bearer Request メッセージの受信時に、サブスクライバが IDLE 状態に移行している場合の Release Access Bearer Request および Response メッセージ。

最適化された GTPv2 エンコーダおよびデコーダは、次のメッセージに対して提供されます。

- ベアラーリソースコマンド
- Change Notification Request と Response
- Create Bearer Request および Response
- Create IDFT Request と Response
- Create Session Request と Response
- ベアラー削除コマンド
- Delete Bearer Failure Indication
- Delete Bearer Request および Response
- Delete IDFT Request と Response
- Delete Session Request と Response
- Downlink Data Notification Acknowledgment
- Downlink Data Notification Failure Indication
- Download Datalink Notification Request
- エコー要求およびエコー応答
- Modify Bearer Request および Response
- ベアラー変更コマンド
- Modify Bearer Failure Indication
- Update Bearer Request および Response

機能設定

S11、S5、および S5e インターフェイスでこの機能を構成するには、次の構成を使用します。

```
config
   instance instance-id instance_id
      endpoint gtp
         replicas replica_count
         vip-ip ipv4_address  vip-port ipv4_port_number
         vip-ipv6 ipv6_address  vip-ipv6-port ipv6_port_number
         dual-stack-transport { true | false }
         enable-go-encdec { true | false }
         interface interface_name
            enable-go-encdec { true | false }
            end
```

注：

- **enable-go-encdec { true | false }**：インターフェイスの Go 言語ベースの GTPv2 エンコーダおよびデコーダを有効にします。
- **dual-stack-transport { true | false }**：IPv6 または IPv4 アドレスを指定できるデュアルスタック機能を有効にします。この機能を有効にするには、true を指定します。

設定例

次に、設定例を示します。

```
config
    instance instance-id 1
       endpoint gtp
          replica 2
          vip-ip 209.165.200.224 vip-port 2022
          vip-ipv6 ipv6_address 2001:db8:1::2 22
          dual-stack-transport true
          enable-go-encdec false
          exit
       interface s5e
          replica 3
          vip-ip 209.165.200.225 vip-port 2022
          vip-ipv6 ipv6_address 2001:db8:1::3 22
          dual-stack-transport true
          enable-go-encdec true
          exit
       interface s11
          replica 3
          vip-ip 209.165.200.226 vip-port 2022
          vip-ipv6 ipv6_address 2001:db8:1::4 22
          dual-stack-transport true
          enable-go-encdec true
          exit
       interface s5
          replica 3
          vip-ip 209.165.200.227 vip-port 2022
          vip-ipv6 ipv6_address 2001:db8:1::5 22
          dual-stack-transport true
          enable-go-encdec true
          end
```

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

最適化された GTPv2 エンコーダおよびデコーダ機能では、次の統計情報がサポートされています。

Go ベースのエンコーダデコーダの Grafana クエリ：

クエリ：

```
sum(irate(gtpc_golang_enc_dec_stats{namespace="$namespace"}[60s])) by
(gtpc_msg_type, gtpc_msg_operation, gtpc_msg_status)
```

凡例：

```
{{gtpc_msg_type}}-{{gtpc_msg_operation}}-{{gtpc_msg_status}}
```

GR スプリットを使用した GTPC エンドポイント

機能説明

GR スプリット機能により、スケーリングされた GTP トラフィックの処理が可能になり、CPU の最適な使用が促進されます。GR スプリット機能は、GTPC-EP の複数のアクティブインスタンスを起動し、GR インスタンスに基づいてトラフィックの分割を実行します。これは、UDP プロキシバイパス機能を支援する上で役立ちます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

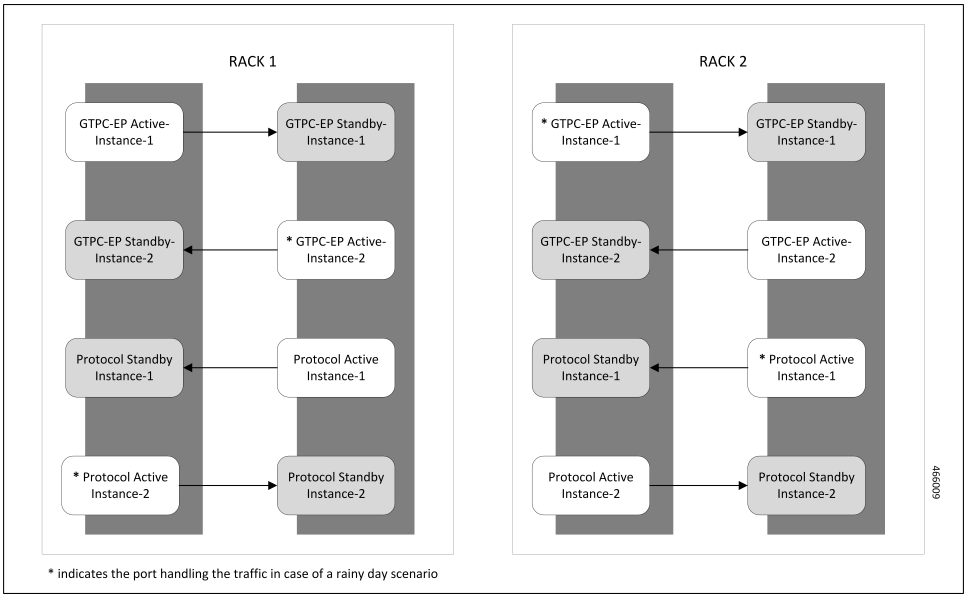
UDP プロキシマージモードがこの機能の前提条件です。

- 正常時のシナリオの場合、RACK1（GTPC-EP Active Instance-1）と RACK2（GTPC-EP Active Instance-2）は、GR-Instance-1 と GR Instance-2 からのトラフィックをそれぞれ処理します。
- 障害時のシナリオでは、GR インスタンス-1（S11、S5E、S5）および GR インスタンス-2（S11、S5E、S5）のトラフィックが 2 つの GTPC-EP インスタンス間で分割されます。

  障害時のシナリオでは、RACK2 がダウンしていると仮定すると、RACK1 は GR Instance-1 と GR Instance-2 のすべてのトラフィックを処理します。GR スプリットの実装では、GTPC-EP Active Instance-1 は GR Instance-1 のすべてのインターフェイスの GTP トラフィッ

クを処理し、GTPC-EP Active Instance-2 は GR Instance-2 のすべての GTP トラフィックを処理します。

図 *99* : マージモードおよび *GR* 分割の *GTPC-EP*



S11 と S5 による GTPC エンドポイント インターフェイス分割

機能説明

GTPC インターフェイス分割機能を使用すると、インターフェイスに基づいて GTPC エンドポイントを分割できます。cnSGW-C は、GTPC ポッドを 2 つのインターフェイスである S11 と S5 に分割します。これにより、S11（SGW-Ingress）、S5E（SGW-Egress）、および S5（SMF-Ingress）のすべての GTPC 着信および発信トラフィックの処理が可能になります。

機能の仕組み

ここでは、この機能の仕組みを説明します。

GR インスタンス分割モードまたは UDP プロキシマージモードであることがこの機能の前提条件です。

この機能はデフォルトで無効に設定されています。この機能を設定する際、S11 および S5 GTPC エンドポイントに個別の内部および外部 VIP アドレスを指定し、2 つの GTPC-EP ポッドを展開します。

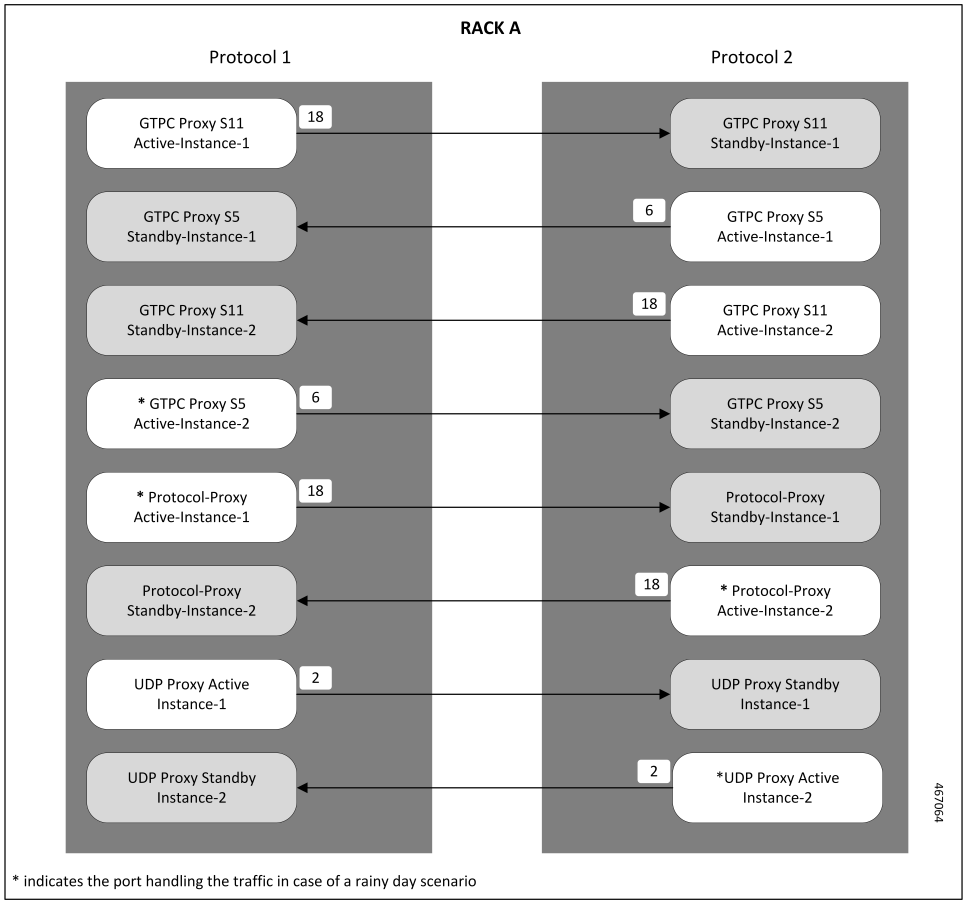
機能を設定すると、cnSGW-Cは、GTPCインターフェイスとGRインスタンスIDに基づいて、SMF サービス、SGW サービス、およびノードマネージャから GTPC ポッドに IPC コールを送信します。

次に、GTPC インターフェイス分割機能の推奨事項を示します。

- CPU コア：
  - S5 インターフェイス：6
  - S11 インターフェイス：18
  - UDP プロキシ：2
- VIP 設定：S5 および S11 インターフェイス用の個別の内部および外部 VIP
- ディスパッチャの構成は、アップグレード前の既存の構成と同じにする必要があります。

次の図は、GTPC インターフェイス分割ポッドのレイアウトを示しています。

図 *100 : GTPC* インターフェイス分割ポッドレイアウト



機能設定

この機能を設定するには、次の設定を使用します。

```
config
   instance instance-id instance_id
      endpoint gtp
      interface s11
         standalone true
         cpu max-process cpu_core_value
         internal-vip internal_vip_address
         vip-ip ipv4_address vip-port ipv4_port_number
         vip-ipv6 ipv6_address vip-ipv6-port ipv6_port_number
         dual-stack-transport { true | false }
         vip-interface vip_interface_value
         exit
      exit
      interface s5
         vip-ip ipv4_address vip-port ipv4_port_number
         vip-ipv6 ipv6_address vip-ipv6-port ipv6_port_number
         dual-stack-transport { true | false }
         vip-interface vip_interface_value
         end
```

注：

- **standalone true**：インターフェイス用の個別のポッドを使用してスタンドアロンモードで実行するようにインターフェイスを構成します。
- **cpu max-process** *cpu_core_value*：インターフェイスの CPU の CPU コア値を指定します。これにより、ポッドの GO_MAX_PROCS パラメータ値が設定されます。
- **dual-stack-transport { true | false }**：IPv6 または IPv4 アドレスを指定できるデュアルスタック機能を有効にします。この機能を有効にするには、true を指定します。

設定例

次に、設定例を示します。

```
config
   instance instance-id 1
      endpoint gtp
      interface s11
         standalone true
         cpu max-process 18
         internal-vip 209.165.200.225
         vip-ip 209.165.200.226 vip-interface bd1.gtp.2131
         exit
      exit
      interface s5
      vip-ip 209.165.200.228 vip-interface bd1.gtp.2131
      end
```

GTPC IPC クロスラックサポート

機能説明

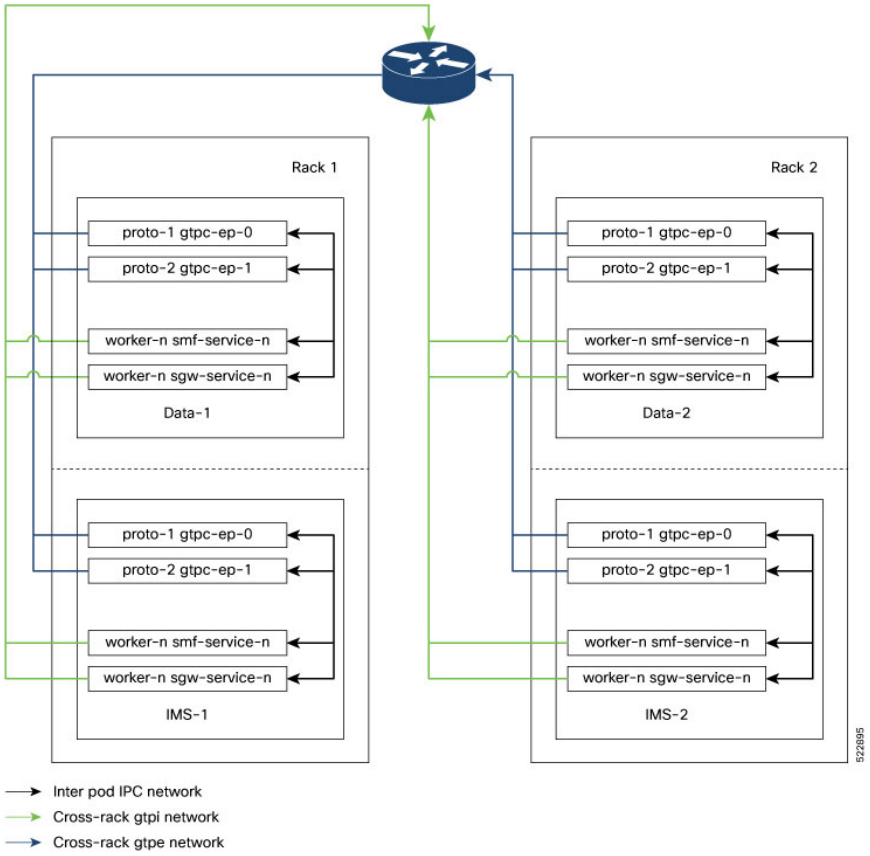
cnSGW-C と SMF を使用して GR セットアップアクティビティを実行する場合、次のシナリオのように、これら 2 つのラック間で GTPC メッセージの処理を最適化できます。

- cnSGW-C から SMF サービスポッドへの IPC メッセージのセットは、`gtpc-ep` ポッドを介して 2 回流れるため、メッセージのエンコードとデコードのオーバーヘッドにつながります。
- GR ペア内で、cnSGW-C や SMF などのサービスポッドが対応するピア GTPC ノードにメッセージを直接ルーティングできれば、これらの IPC メッセージは 1 つ以上の処理ステップを回避できます。

（注） cnSGW または SMF インターフェイスで GTPC IPC を有効にする設定を適用する前に、クラスタ同期を使用してラック間ルーティングネットワークを適用する必要があります。ラックサーバー間で新しいルーティング可能なネットワークをサポートするための BGP ルートを追加するには、より多くの設定が必要です。

次の図は、機能、コア セットアップ アクティビティ、およびそれらの相互接続のサポートに必要な新しいネットワーク レイアウトの設計を表しています。

図 *101 : SGW-GTPC* ラック間 *IPC*



次の手順は、2 つのラック間での GTPC メッセージ処理の最適化とクロスラックエンドポイントの展開のために実行されます。

- IMS-1 Rack-1 の cnSGW-C は、IPC 要求を、クロスラック GTPI ネットワークを介してルータに渡す DATA-2 Rack-2 の PGW GTPC-EP に内部的にルーティングします。
- ルータは、要求を `gtpc-ep` ポッドに転送するためのネクストホップとして GTPE ネットワークを使用します。
- GTPI ネットワークと GTPE ネットワークは、展開のプロセス中にラックに追加される新しいネットワークです。
- また、この機能には、アクティブな `gtpc-ep` ポッドで受信される内部 GTPC IPC メッセージが必要です。
- このプロセスでは、cnSGW-C から SMF サービスポッドへの IPC メッセージが GTPC-EP ポッドを介して流れ、メッセージのエンコードとデコードの労力が 2 発生します。
- GR ペア内で、これらのサービスポッド（cnSGW-C および SMF）が対応するピア GTPC ノードにメッセージを直接ルーティングできれば、このような IPC メッセージは余分な処理を 1 つ回避できます。

（注） 設定されたプロトコルノードは、S5 および S5e VIP グループが展開されているのと同じ VIP グループにある必要があります。

機能の仕組み

ここでは、この機能の仕組みを説明します。

- `SMF-Ops-Center` では、IMS ラックおよびデータラックの各ラックに `GTPC-EP Geo` エンドポイントを設定できます。
- `SMF-Ops-Center` CLI では、IMS ラックとデータラックの各ラックに `GTPC-EP Geo` エンドポイントを設定できます。

（注） GTPC メッセージの最適化は、4 つのすべてのインスタンス、またはこれらのラック内の GTPC エンドポイントのインスタンスのサブセットに適用できます。SMF および cnSGW-C サービスポッドが IPC インターフェイスを介して GTPv2 メッセージをルーティングできる IP アドレスのリストを設定するために、新しい要素が GTPC エンドポイントの下に追加されます。

ラック間 GTPC IPC のアップグレードと有効化

ここでは、ラック間 GTPC IPC をアップグレードして有効にする方法について説明します。

cnSGW-C および SMF インターフェイスで GTPC IPC 最適化を設定、アップグレード、および有効にする前に、次の手順を実行する必要があります。

- ラック間ルーティングは、ブランチのコアネットワークに適用する必要があります。
- ラックサーバー全体で新しいルーティング可能なネットワークをサポートするための追加ルートが加わると、GTPC、cnSGW-C、SMF など、より多くの設定パラメータが有効になります。

（注） 設定に関するポイントは次のとおりです。

- S5 出力 MBR 要求の SGW サービスポッドは、リスト内の PGWc および SMF IP への IPC メッセージに使用されます。
- S5 CBR、UBR、および DBR 要求の SMF サービスポッドは、リスト内の cnSGW-C への IPC メッセージに使用されます。
- IPC メッセージは、ピアノードでタイムアウトが発生するたびに、メッセージを再試行するためにそれぞれのインターフェイスで N3 または T3 設定を再利用します。

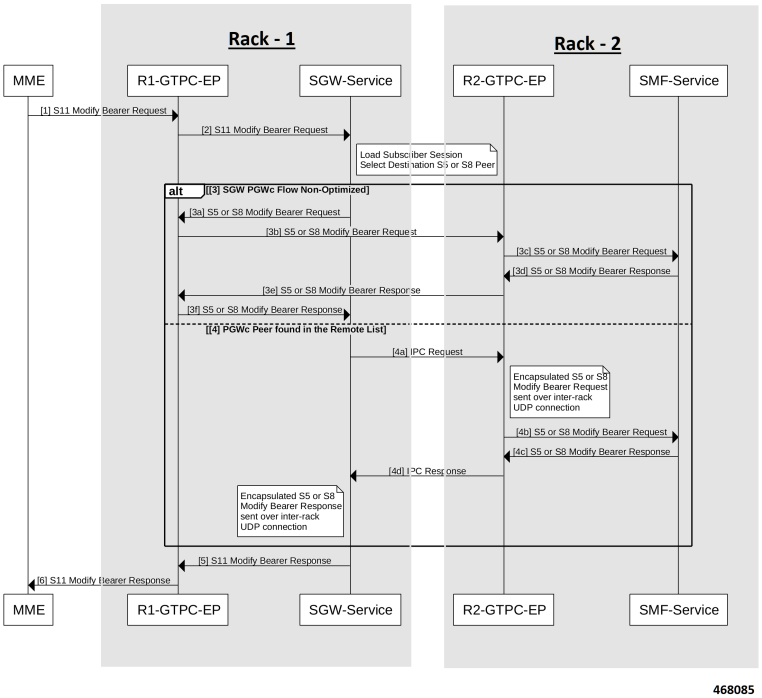
コール フロー

ここでは、この機能の主要なコールフローについて説明します。

ラック間 IPC コールフローを搭載した cnSGW-C GTPC 最適化

ここでは、ラック間 IPC による cnSGW-C GTPC 最適化コールフローについて説明します。

図 *102 :* ラック間 *IPC* コールフローを搭載した *cnSGW-C GTPC* 最適化



468085

表 *184 :* ラック間 *IPC* コールフローを搭載した *cnSGW-C GTPC* 最適化コールフローの説明

ステップ	説明
1	MME が S11 ベアラー変更要求を R1-GTPC-EP に送信します。
2	R1-GTPC-EP が S11 ベアラー変更要求を SGW サービスに送信します。 （注） SGW サービスセクションでは、次の機能が実行されます。 • サブスクライバセッションのロード。 • 宛先を S5 ピアまたは S8 ピアとして選択。
3	次に、Alt SGW PGWc フローの最適化されていないシナリオのサブステップを示します。
3a	SGW サービスは、S5 または S8 ベアラー変更要求を R1-GTPC-EP に送信します。
3b	R1-GTPC-EP が、S5 または S8 ベアラー変更要求を R2-GTPC-EP に送信します。
3c	R2-GTPC-EP が、S5 または S8 ベアラー変更要求を SMF サービスに送信します。
3d	SMF サービスが、S5 または S8 ベアラー変更応答を処理して R2-GTPC-EP に送信します。
3e	R2-GTPC-EP が、R1-GTPC-EP にベアラー変更応答を送信します。
3f	R1-GTPC-EP が、SGW サービスにベアラー変更応答を送信します。
4	または、PGWc ピアシナリオをリモートリストで使用できます。 次に、このシナリオのサブステップを示します。
4a	SGW サービスが IPC 要求を R2-GTPC-EP に送信します。 （注） R2-GTPC-EP セクションで、次の機能が実行されます。 • S5 または S8 ベアラー変更要求のカプセル化。 • ラック間 UDP 接続経由の送信。
4b	R2-GTPC-EP が、S5 または S8 ベアラー変更要求を SMF サービスに送信します。
4c	SMF サービスが、S5 または S8 ベアラー変更応答を処理して R2-GTPC-EP に送信します。

ステップ	説明
4d	R2-GTPC-EP が IPC 応答を SGW サービスに送信します。 （注） SGW サービスセクションでは、次の機能が実行されます。 • S5 または S8 ベアラー変更応答のカプセル化。 • ラック間 UDP 接続経由の送信。
5	SGW サービスが、R1-GTPC-EP にベアラー変更応答を送信します。
6	R1-GTPC-EP が、MME にベアラー変更応答を送信します。

ラック間 IPC による SMF および PGWc GTPC 最適化コールフロー

ここでは、ラック間 IPC による SMF および PGWc GTPC 最適化コールフローについて説明します。

図 *103 :* ラック間 *IPC* による *SMF* および *PGWc GTPC* 最適化コールフロー

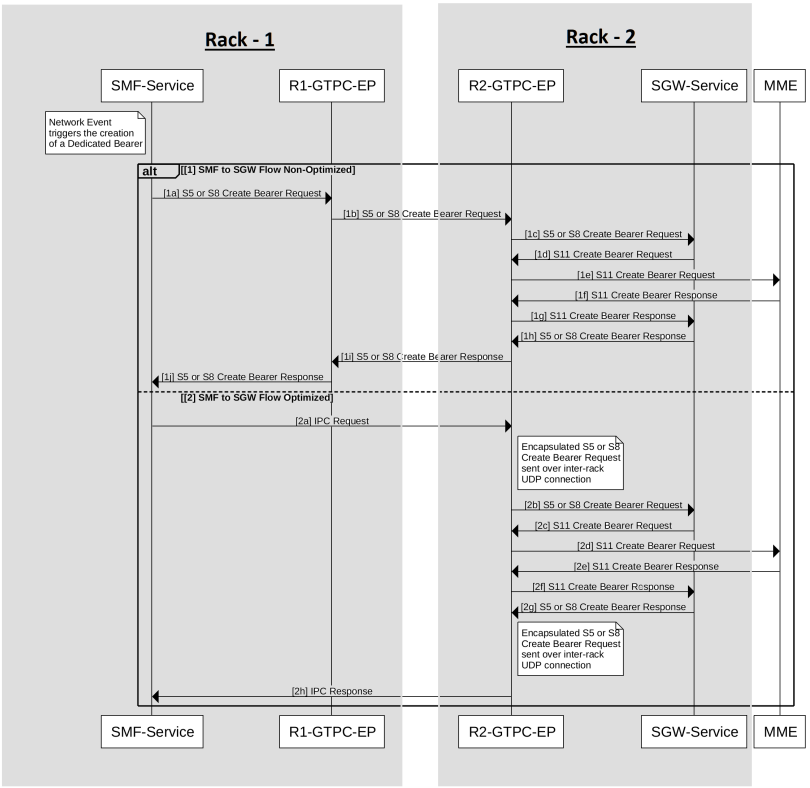


表 *185 :* ラック間 *IPC* による *SMF* および *PGWc GTPC* 最適化コールフローの説明

ステップ	説明
1	次に、Alt SMF から SGW へのフローが最適化されていないシナリオのサブステップを示します。 （注） SMF サービスセクションで、次の手順が実行されます。 • ネットワーキングイベントのトリガー。 • リゾルトベアラーの作成。
1a	SMF サービスが S5 または S8 Create Bearer Request を R1-GTPC-EP に送信します。
1b	R1-GTPC-EP が S5 または S8 Create Bearer Request を R2-GTPC-EP に送信します。
1c	R2-GTPC-EP が S5 または S8 Create Bearer Request を SGW サービスに送信します。

ステップ	説明
1d	SGW サービスが S11 Create Bearer Request を処理して R2-GTPC-EP に送信します。
1e	R2-GTPC-EP が S11 Create Bearer Request を MME に送信します。
1f	MME が S11 Create Bearer Response を処理して R2-GTPC-EP に送信します。
1g	R2-GTPC-EP が S11 Create Bearer Response を SGW サービスに送信します。
1h	SGW サービスが S5 または S8 Create Bearer Response を処理して R2-GTPC-EP に送信します。
1i	R2-GTPC-EP が S5 または S8 Create Bearer Response を R1-GTPC-EP に送信します。
1j	R1-GTPC-EP が S5 または S8 Create Bearer Response を SMF サービスに送信します。
2	SMF から SGW へのフロー最適化シナリオのサブステップは次のとおりです。
2a	SMF サービスが IPC 要求を R2-GTPC-EP に送信します。 （注） R2-GTPC-EP セクションで、次の手順が実行されます。 • S5 または S8 Create Bearer Request のカプセル化。 • ラック間 UDP 接続経由の送信。
2b	R2-GTPC-EP が S5 または S8 Create Bearer Request を SGW サービスに送信します。
2c	SGW サービスが S11 Create Bearer Request を処理して R2-GTPC-EP に送信します。
2d	R2-GTPC-EP が S11 Create Bearer Request を MME に送信します。
2e	MME が S11 Create Bearer Response を処理して R2-GTPC-EP に送信します。
2f	R2-GTPC-EP が S11 Create Bearer Response を SGW サービスに送信します。
2g	SGW サービスが S5 または S8 Create Bearer Response を処理して R2-GTPC-EP に送信します。 （注） R2-GTPC-EP セクションで、次の手順が実行されます。 • S5 または S8 Create Bearer Request のカプセル化。 • ラック間 UDP 接続経由の送信。
2時間	R2-GTPC-EP が IPC 応答を処理して SMF サービスに送信します。

機能設定

この機能を設定するには、次の設定を使用します。

```
config
instance instance_id 1
   endpoint endpoint_name
      interface gtp-inter-rack gtp_inter_rack_name
         vip-ip vip_ip_address vip-port vip_port_address vip-interface
vip_interface_address
         gtpc-ipc gtpc_ipc_name
         gtp-peer-entry gtp_peer_entry_address port port_address
remote-gtp-peer-list remote_gtp_peer_list_addresses
         end
```

（注） 前述の新しい設定例の拡張シナリオは次のとおりです。

- S5 出力 MBR 要求の cnSGW-C サービスポッドでは、前述のリストにある PGWc または SMF IP 宛ての IPC メッセージが使用されます。
- 同様に、S5 CBR、UBR、および DBR 要求の SMF サービスポッドでは、前述のリストにある cnSGW への IPC メッセージが使用されます。
- ピアノードでタイムアウトがある場合、メッセージを再試行するためにそれぞれのインターフェイスの N3 または T3 設定が IPC メッセージで再利用されます。

注：

- **instance** *instance_id 1*：インスタンス ID を指定します。
- **endpoint** *endpoint_name*：エンドポイント名を指定します。
- **gtp-inter-rack** *gtp_inter_rack_name*：インターフェイス名を指定します。選択する **gtp-inter-rack** 名を指定します。これは、クロスラックルーティング用に追加された新しいインターフェイスです。
- **vip-ip** *vip_ip_address* **vip-port** *vip_port_address* **vip-interface** *vip_interface_address*：GTP IPC エンドポイントサーバーの IP である、**vip-ip**GTP IPC エンドポイントサーバーのリスニングポートである、**vip-port**および GTP IPC エンドポイント サーバー インターフェイス VLAN である **vip-interface** のアドレスを指定します。
- **gtpc-ipc** *gtpc_ipc_name*：インターフェイス名を指定します。選択する **gtp-ipc** 名を指定します。
- **gtp-peer-entry** *gtp_peer_entry_address* **port** *port_address* **remote-gtp-peer-list** *remote_gtp_peer_list_addresses*：他のラックまたはインスタンス（複数行）に設定されたリモート GTP IPC ピア IP である、**gtp-peer-entry**リモート GTP IPC ピアポートである、**port** および **remote-gtp-peer-list** のリストのアドレスを指定します。これは、**gtp-peer-entry** に

対応するラックまたはインスタンス上の S5 および S5e リモート GTP ピアエンドポイントのリストです。

設定例

次に、設定例を示します。

```
config
instance instance-id 1
 endpoint GTP
   interface gtp-inter-rack
     vip-ip 209.165.202.130 vip-port 9084 vip-interface bd2.gtpe.2101
     gtpc-ipc
     gtp-peer-entry 209.165.202.131 port 9084 remote-gtp-peer-list [ 209.165.202.140
209.165.202.141 ]
     end
```

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

KPI のサポート

GTPC IPC クロスラックサポート機能では、次の統計情報がサポートされています。

1. KPI 名：**udp_rpc_request_total**

   次の表に、**udp_rpc_request_total** KPI の詳細を示します。

説明	機能	ラベル名	有効な値
この KPI には、UDP クライアント エンドポイント リクエスト メッセージの総数が表示されます。	機能の出力はラック間 RPC 要求の合計です。	MessageName	CBR、UBR、MBR などの gptv2 メッセージ タイプ
		retry	再試行かどうか
		status	要求メッセージのステータス（成功または失敗）。
		status_code	0/1/2

2. KPI 名：**udp_rpc_response_total**

   次の表に、**udp_rpc_response_total** KPI の詳細を示します。

説明	機能	ラベル名	有効な値
この KPI には、UDP クライアントエンドポイントの応答メッセージの総数が表示されます。	機能の出力は、ラック間 RPC 応答の合計です。	MessageName	CBR、UBR、MBR などの gptv2 メッセージタイプ
		status	応答メッセージのステータス（成功または失敗）。
		status_code	0/1/2

（注） 現在の実装では、サービス ポッドに存在するため、クライアント側でのみ KPI がサポートされており、パフォーマンスに影響を与えることなく KPI を有効にできます。

サービス間ポッド通信

機能説明

サブスクライバ用に選択された IMS PDN SGW サービスと SMF サービスが同じクラスタおよび同じラック上にある場合、SGW サービスが SMF サービスにメッセージを送信すると、次のメッセージフローが発生します。

- メッセージは、S5e gtpc-ep インターフェイスからネットワーク インターフェイスに送信されます。
- メッセージは、gtpc-ep から SMF サービスへの S5 インターフェイスに戻ります。

併置されているサブスクライバの場合、通信は SGW サービスと SMF サービスの間で行われます。このアプローチにより、gtpc-ep の処理負荷が軽減されます。

（注） SGW サービスと SMF サービス間の直接通信は、モニタープロトコルとモニターサブスクライバのメッセージ転送ではサポートされていません。

機能の仕組み

ここでは、この機能の仕組みを説明します。

sgw-service は、次の要求を処理するために smf-service と通信します。

- セッション作成要求

- ベアラー変更要求
- セッション削除要求

smf-service は、次の要求を処理するために sgw-service と通信します。

- ベアラー作成要求
- ベアラー更新要求
- ベアラー削除要求

sgw-service は、gtpc-ep を介して SMF にベアラー変更コマンドおよびベアラー削除コマンドメッセージを送信します。ベアラー更新要求およびベアラー削除要求がトリガーされると、コマンドメッセージが gtpc-ep を介して sgw-service に送信されます。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

SGW サービスから SMF サービスへの構成コールフローを使用したコラプストコールアタッチ

このセクションでは、SGW サービスから SMF サービスへの構成コールフローを使用したコラプストコールアタッチについて説明します。

図 *104 : SGW* サービスから *SMF* サービスへの構成コールフローを使用したコラプストコールアタッチ

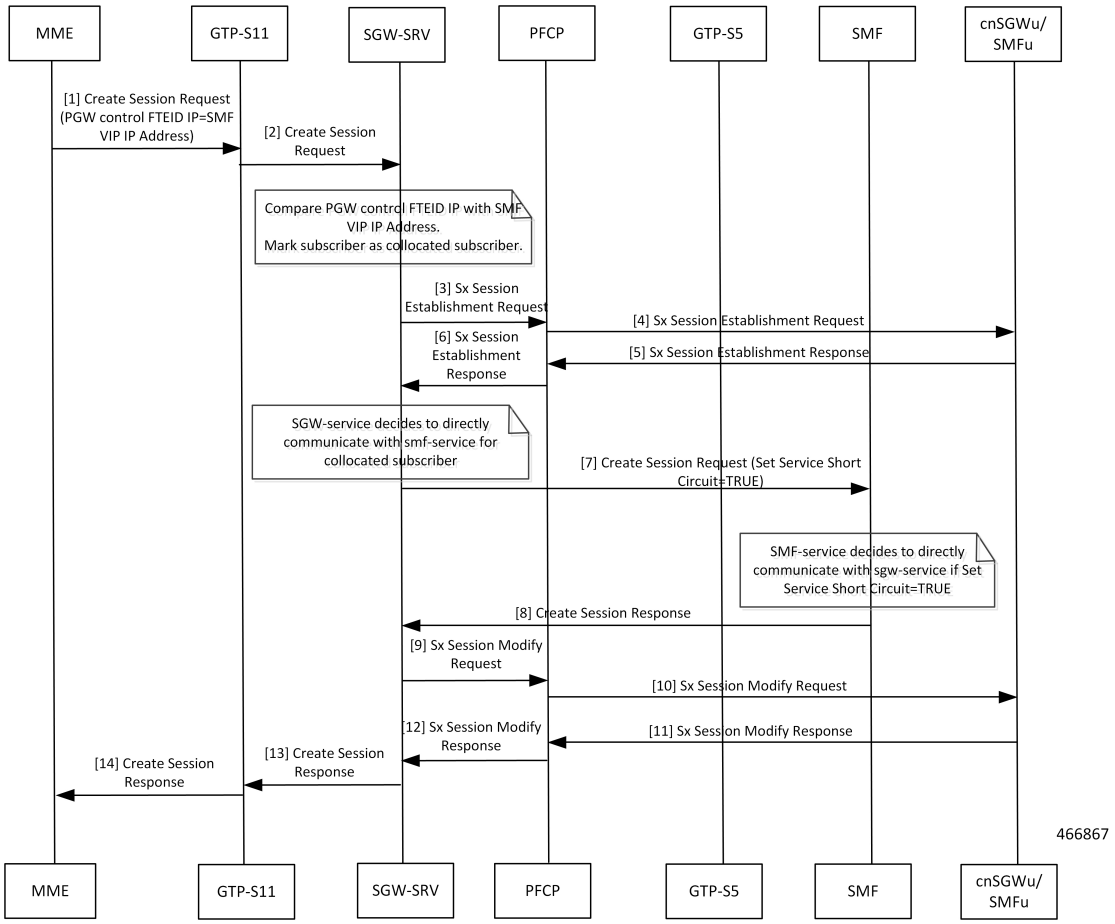


表 *186 : SGW* サービスから *SMF* サービスへの構成コールフローを使用したコラプストコールアタッチの説明

ステップ	説明
1	MME がセッション作成要求を GTP-S11 に送信します。
2	GTP-S11 がセッション作成要求を SGW-SRV に送信します。
3	SGW-SRV が Sx セッション確立要求を PFCP に送信します。
4	PFCP が Sx セッション確立要求を cnSGWu/SMFu に送信します。
5	cnSGWu/SMFu が Sx セッション確立応答を PFCP に送信します。
6	PFCP が Sx セッション確立応答を SGW-SRV に送信します。
7	SGW-SRV がセッション作成応答を SMF に送信します。サービスショート回線は TRUE に設定されます。

ステップ	説明
8	SMF がセッション作成応答を SGW-SRV に送信します。
9	SGW-SRV がSx セッション変更要求を PFCP に 送信します。
10	PFCP が Sx セッション変更要求を cnSGWu/SMFu に送信します。
11	cnSGWu/SMFu がSx セッション変更応答を PFCP に送信します。
12	PFCP が Sx セッション変更応答を SGW-SRV に送信します。
13	SGW-SRV がセッション作成応答を GTP-S11 に送信します。
14	GTP-S11 がセッション作成応答を MME に送信します。 併置されたサブスクライバの場合、SGW-SRV は次を受信します。 • ベアラー要求を変更し、MME からセッション要求を削除すると、SGW-SRV はこれらのメッセージを SMF サービスに伝達します。 • SMF からベアラー作成要求、ベアラー更新要求、およびベアラー削除要求を行うと、SGW-SRVはこれらの要求メッセージに対する応答メッセージを SMF サービスに伝達します。

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

統計のサポート

SMF サービスに直接送信されるメッセージを確認するには、次に示すように sgw_service_stats クエリに svc_to_svc フィールドを追加します。

クエリ：sum(irate(sgw_service_stats(status=~"attempted")[30s]) by (sgw_procedure_type,status, interface, svc_to_svc)

凡例：{{interface}} -> {{sgw_procedure_type}} | {{svc_to_svc}} | {{status}}

MBR コールフロー最適化

機能説明

cnSGW-C は、Modify Bearer Request および Modify Bearer Response（MBR）コールフローの最適化をサポートし、I/O 操作の削減、トランザクションの待機時間の短縮、およびマルチ PDN シナリオでのパフォーマンスの向上を実現します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

次の機能は、MBR コールフローの最適化について説明します。

- I/O 操作を削減するために、cnSGW-C は、SGW サービスに対するすべてのベアラー変更要求を単一の GRPC コールに結合し、SGW サービスポッドからプロトコルポッドへの Sx 変更要求を単一の GRPC コールに結合します。
- GTPC-EP でのトランザクション待機時間を短縮するために、cnSGW-C は、ベアラー変更要求の受信後、直ちに GTPC-EP からベアラー変更応答を送信します（最後の MBR を除く）。

  GTPC-EP が、単一のベアラー変更リクエストリストメッセージにすべてのベアラー変更要求を組み合わせ、SGW サービスに送信します。
- SGW-service は、すべてのベアラー変更応答を単一のベアラー応答の変更リストメッセージにまとめ、GTPC-EP に送信します。

  SGW サービスは、UPF へのすべての Sx 変更要求を単一の Sx 変更要求リストメッセージに結合し、プロトコルポッドに送信します。プロトコルポッドは、個々の Sx 変更要求を UPF に送信します。
- プロトコルポッドは UPF からのすべての Sx 変更応答を待ち、それらを単一の Sx 変更応答リストに結合して SGW サービスに送信します。
- 非マージモードでは、UDP プロキシはローカル TEID とリモート TEID キャッシュ情報を維持します。マージモードでは、GTPC-EP はローカル TEID とリモート TEID のキャッシュ情報を維持します。
- GTPC-EP が受信したベアラー変更要求 TEID キャッシュエントリを見つけられない場合、ベアラー変更要求はすぐに SGW サービスに転送されます。

  予想されるすべてのベアラー変更要求が MBR キャッシュの有効期限内に受信されない場合、受信されたベアラー変更要求のみが SGW サービスに送信されます。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

MME 間 HO を使用したアイドル-アクティブ移行のコールフロー

ここでは、eNodeB から gNodeB に移動するための MME 間ハンドオーバー（NSA MBR）を使用した現在のコールフローについて説明します。

図 *105 : MME* 間 *HO* を使用したアイドル-アクティブ移行のコールフロー

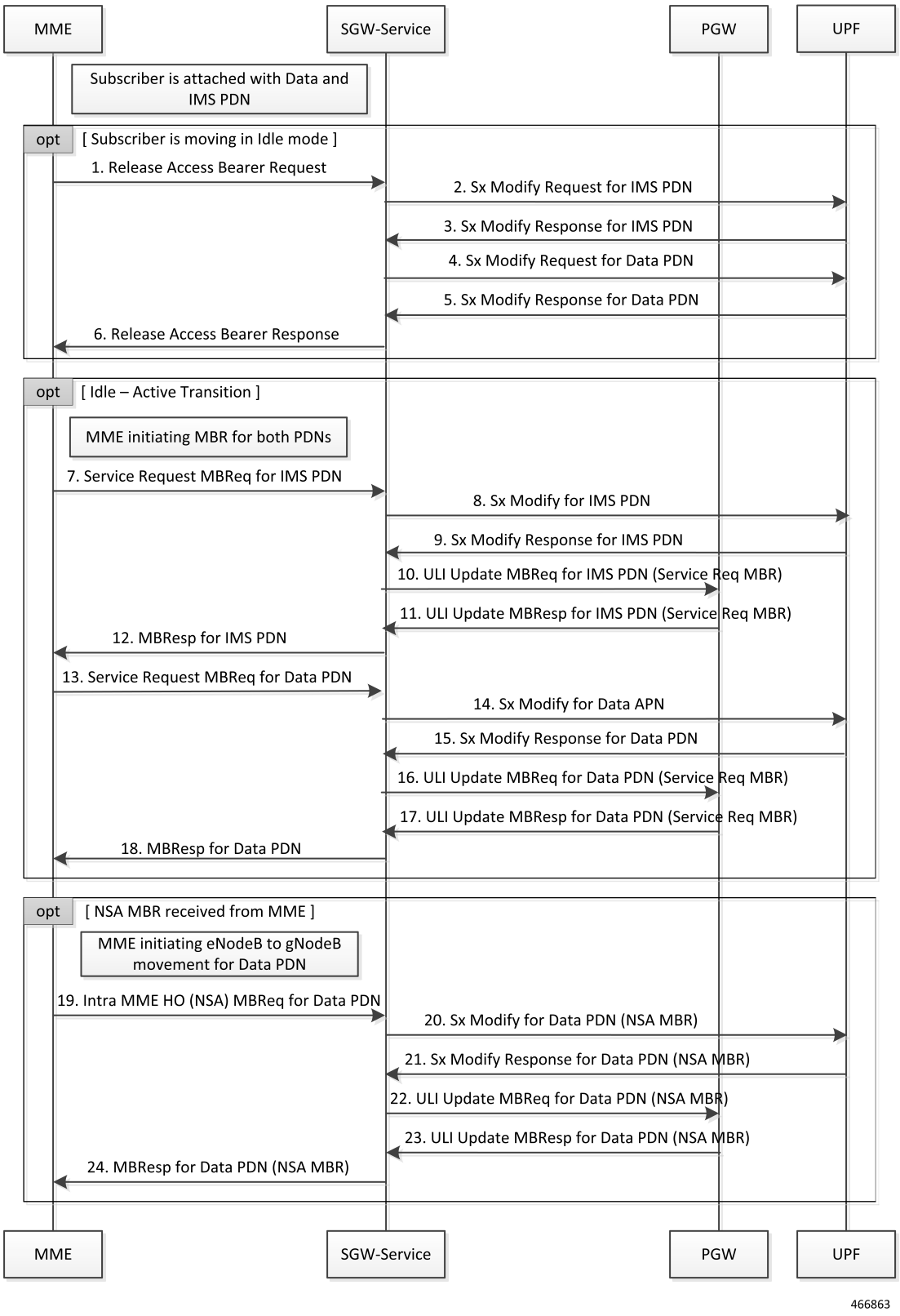

MME
SGW-Service
PGW
UPF
Subscriber is attached with Data and IMS PDN
opt
[Subscriber is moving in Idle mode]
1. Release Access Bearer Request
2. Sx Modify Request for IMS PDN
3. Sx Modify Response for IMS PDN
4. Sx Modify Request for Data PDN
5. Sx Modify Response for Data PDN
6. Release Access Bearer Response
opt
[Idle – Active Transition]
MME initiating MBR for both PDNs
7. Service Request MBReq for IMS PDN
8. Sx Modify for IMS PDN
9. Sx Modify Response for IMS PDN
10. ULI Update MBReq for IMS PDN (Service Req MBR)
11. ULI Update MBResp for IMS PDN (Service Req MBR)
12. MBResp for IMS PDN
13. Service Request MBReq for Data PDN
14. Sx Modify for Data APN
15. Sx Modify Response for Data PDN
16. ULI Update MBReq for Data PDN (Service Req MBR)
17. ULI Update MBResp for Data PDN (Service Req MBR)
18. MBResp for Data PDN
opt
[NSA MBR received from MME]
MME initiating eNodeB to gNodeB movement for Data PDN
19. Intra MME HO (NSA) MBReq for Data PDN
20. Sx Modify for Data PDN (NSA MBR)
21. Sx Modify Response for Data PDN (NSA MBR)
22. ULI Update MBReq for Data PDN (NSA MBR)
23. ULI Update MBResp for Data PDN (NSA MBR)
24. MBResp for Data PDN (NSA MBR)
MME
SGW-Service
PGW
UPF
466863

表 *187 : MME* 間 *HO* を使用したアイドル-アクティブ移行のコールフローの説明

ステップ	説明
1	MME が SGW サービスにアクセスベアラーのリリース（RAB）要求を送信します。
2	SGW サービスが IMS PDN の Sx 変更要求を UPF に送信します。
3	UPF が IMS PDN の Sx 変更応答を SGW サービスに送信します。
4	SGW サービスがデータ PDN の Sx 変更要求を UPF に送信します。
5	UPF がデータ PDN の Sx 変更応答を SGW サービスに送信します。
6	SGW サービスが MME にアクセスベアラーのリリース応答メッセージを送信します。
7	MME が IMS PDN のサービス要求 MBRequest を SGW サービスに送信します。
8	SGW サービスが IMS PDN の Sx 変更要求を UPF に送信します。
9	UPF が IMS PDN の Sx 変更応答を SGW サービスに送信します。
10	SGW サービスが、IMS PDN（サービスリクエスト MBR）の ULI Update MBRequest を PGW に送信します。
11	PGW が、IMS PDN（サービス要求 MBR）の ULI Update MBResponse を SGW サービスに送信します。
12	SGW サービスが IMS PDN の MBResponse を MME に送信します。
13	MME がデータ PDN のサービス要求 MBRequest を SGW サービスに送信します。
14	SGW サービスがデータ PDN の Sx 変更要求を UPF に送信します。
15	UPF がデータ PDN の Sx 変更応答を SGW サービスに送信します。
16	SGW サービスが、データ PDN（サービスリクエスト MBR）の ULI Update MBRequest を PGW に送信します。
17	PGW が、データ PDN（サービス要求 MBR）の ULI Update MBResponse を SGW サービスに送信します。
18	SGW サービスが、データ PDN の MBResponse を MME に送信します。
19	MME が、データ PDN のイントラ MME HO（NSA）MBRequest を SGW サービスに送信します。
20	SGW サービスがデータ PDN（NSA MBR）の Sx 変更要求を UPF に送信します。
21	UPF がデータ PDN（NSA MBR）の Sx 変更応答を SGW サービスに送信します。

ステップ	説明
22	SGW サービスがデータ PDN（NSA MBR）の ULI Update MBRequest を UPF に送信します。
23	UPF がデータ PDN の ULI Update MBResponse を SGW サービスに送信します。
24	SGW サービスがデータ PDN（NSA MBR）の MBResponse を MME に送信します。

MBR 最適化コールフロー

ここでは、アイドルからアクティブに移行するための MBR 最適化の高度なコールフローについて説明します。

図 *106 : MBR* 最適化コールフロー

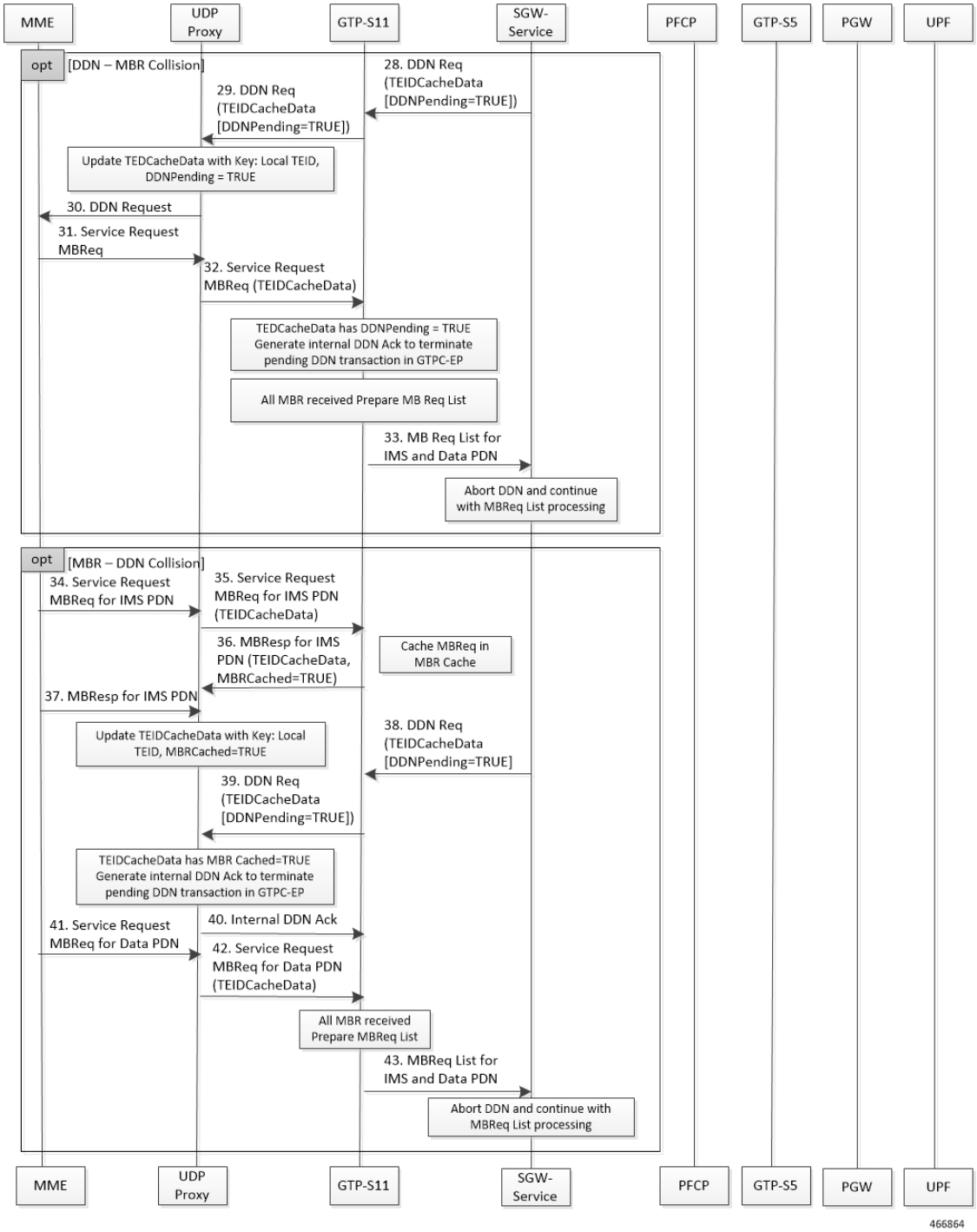


表 *188 : MBR* 最適化コールフローの説明

ステップ	説明
1	MME が Release Access Bearer（RAB）要求を UDP プロキシに送信します。
2	UDP プロキシが RAB 要求を GTP-S11 に転送します。
3	GTP-S11 が RAB 要求を SGW サービスに転送します。
4	SGW サービスがすべての PDN の Sx Modify List を PFCP に送信します。
5	SGW サービスが RAB 応答に応答します。
6	GTP-S11 が RAB 応答を UDP プロキシに送信します。
7	UDP プロキシが RAB 応答を MME に送信します。
8	PFCP が UPF との間で IMS APN の Sx Modify を送受信します。
9	PFCP が UPF との間でデータ APN の Sx Modify を送受信します。
10	PFCP が、すべての PDN の Sx Modify Request List を SGW サービスに送信します。
11	MME が IMS PDN のサービスリクエスト MBRequest を UDP プロキシに送信します。
12	UDP プロキシが、TEID キャッシュデータを含む IMS PDN のサービスリクエスト MBRequest を GTP-S11 に転送します。
13	GTP-S11 が IMS PDN の MBResponse を UDP プロキシに送信します。
14	UDP プロキシが IMS PDN の MBResponse を MME に転送します。
15	MME がデータ PDN のサービスリクエスト MBRequest を UDP プロキシに送信します。
16	UDP プロキシが、TEID キャッシュデータを含むデータ PDN のサービスリクエスト MBRequest を GTP-S11 に送信します。
17	GTP-S11 が、IMS およびデータ PDN の MBRequest List を SGW サービスに送信します。
18	SGW サービスが IMS およびデータ PDN の Sx Modify List を PFCP に送信します。
19	PFCP が UPF との間で IMS PDN の Sx Modify を送受信します。
20	PFCP が UPF との間でデータ PDN の Sx Modify を送受信します。
21	PFCP が IMS およびデータ PDN の Sx Modify Response List を SGW サービスに送信します。

ステップ	説明
22	SGW サービスが IMS およびデータ PDN（非同期通知）の ULI Update MBRequest List を GTP-S5 に送信します。
23	SGW サービスがデータ PDN の MBResponse List を GTP-S11 に送信します。
24	GTP-S11 がデータ PDN の MBResponse を UDP プロキシに送信します。
25	UDP プロキシがデータ PDN の MBResponse を MME に転送します。
26	GTP-S5 が PGW との間で IMS PDN の ULI Update MBRequest を送受信します。
27	GTP-S5 が PGW との間でデータ PDN の ULI Update MBRequest を送受信します。
28	SGW サービスが DDN 要求を GTP-S11 に送信します。
29	S-11 が DDN 要求を UDP プロキシに転送します。
30	UDP プロキシが DDN 要求を MME に転送します。
31	MME がサービスリクエスト MBRequest を UDP プロキシに送信します。
32	UDP プロキシが、TEID キャッシュデータを含むサービスリクエスト MBRequest を GTP-S11 に送信します。
33	GTP-S11 が IMS およびデータ PDN の MBResponse List を SGW サービスに送信します。
34	MME が IMS PDN のサービスリクエスト MBRequest を UDP プロキシに送信します。
35	UDP プロキシが、TEID キャッシュデータを含む IMS PDN のサービスリクエスト MBRequest を GTP-S11 に送信します。
36	GTP-S11 が IMS PDN の MBResponse を UDP プロキシに送信します。
37	UDP プロキシが IMS PDN の MBResponse を MME に送信します。
38	SGW サービスが DDN 要求を GTP-S11 に送信します。
39	GTP-S11 が DDN 要求を UDP プロキシに転送します。
40	UDP プロキシが内部 DDN 確認応答を GTP-S11 に送信します。
41	MME がデータ PDN のサービスリクエスト MBRequest を UDP プロキシに送信します。
42	MME が TEID キャッシュデータを含むデータ PDN のサービスリクエスト MBRequest を GTP-S11 に送信します。

ステップ	説明
43	GTP-S11 が、IMS およびデータ PDN の MBRequest List を SGW サービスに送信します。

機能設定

この機能を設定するには、次の設定例を使用します。

```
config
   instance istance-id instance_id
      endpoint endpoint_name
         mbr-optimization [ enable { false | true } | mbr-cache-expiry
mbr_cache | teid-cache-expiry teid_cache ]
         exit
      exit
   exit
```

注：

- **mbr-optimization [enable { false | true } | mbr-cache-expiry** *mbr_cache* **| teid-cache-expiry** *teid_cache* **]**：MBR 最適化構成を指定します。
  - **enable { false | true }**：MBR 最適化を有効または無効にします。デフォルトで、ディセーブルになっています。
  - **mbr-cache-expiry** *mbr_cache* ：MBR キャッシュの有効期限の間隔を 1 ミリ秒から 6 秒の整数で指定します（ミリ秒単位）。デフォルト：50 ミリ秒。

    mbr-cache-expiry の値は実行時に変更できることに注意してください。
  - **teid-cache-expiry** *teid_cache* ：TEID キャッシュの有効期限の間隔を、1000 ミリ秒から 1 時間の整数で指定します（ミリ秒単位）。デフォルト：120000 ミリ秒。

設定例

次に、設定例を示します。

```
config
   instance instance-id 1
      endpoint gtp
         mbr-optimization enable true mbr-cache-expiry 60 teid-cache-expiry 180000
         end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config instance instance-id 1 endpoint gtp
instance instance-id 1
endpoint gtp
  replicas               1
```

```
nodes                    1
mbr-optimization
 enable             true
 teid-cache-expiry 180000
 mbr-cache-expiry  60
exit
enable-cpu-optimization true
……
```

OAM サポート

ここでは、MBR最適化機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

MBR 最適化機能では、次の統計情報がサポートされています。

SGW サービスは、GTPC-EP からのベアラー変更要求リストを処理するために *service_request_list* 手順タイプをサポートします。

クエリ：`sum(rate(sgw_service_stats{namespace=~"$namespace", interface="interface_sgw_ingress",sgw_procedure_type="service_request_list", status="attempted"}[1m]))`

判例： `IDLE -> ACTIVE (List)`

MBR 短絡統計は、SGW サービスに送信されたベアラー変更要求リストの統計情報をキャプチャするように強化されています。

クエリ：`sum(irate(gtpc_msg_short_circuit_stats{namespace=~"$namespace"}[60s])) by (gtpc_msg_type, gtpc_short_circuit_category)`

判例：`{{gtpc_msg_type}}, {{gtpc_short_circuit_category}}`

この機能では、次の統計がサポートされます。

- RxModifyBearerReq、SendSCMBResp：GTPC-EP から送信された早期 MB 応答の数。
- RxModifyBearerReq、SendMBReqListToSrv：SGW サービスに送信された MB 要求リストの数。
- RxModifyBearerReq、SendMBReqToSrv：SGW サービスに送信された MB 要求の数。
- RxModifyBearerReq、MBREventExpired：MBR キャッシュで期限切れになった MB 要求の数。

メンテナンス モード

機能説明

メンテナンスモード機能により、GR セットアップ内のクラスタは、サービスを中断することなく、インサービスアップグレード（ローリングアップグレード）を実行できます。メンテナンスモードは、対応するクラスタへのトラフィックのルーティングと連携して、ソースクラスタの役割を実行します。

機能の仕組み

メンテナンスモードフラグがtrueに設定されている場合、クラスタロールが変更され、ラックに対してGRがトリガーされます。スタンバイクラスタは、メンテナンスモードのクラスタの役割を引き継ぎます。この間、モニタリングスレッドはフラグのランタイム値をチェックし、メンテナンスモードフラグがtrueに設定されている場合は実行を一時停止します。デフォルトでは、新規インストールの場合、フラグは false に設定されます。

ソースクラスタとスタンバイクラスタ（ラック）の両方を同時にメンテナンスモードにすることができます。状態に関係なく、ラックサーバーのメンテナンスモードを有効にできます。

従属クラスタに対してメンテナンス手順を実行中の場合、システムをメンテナンスモードに移行させることができます。メンテナンス作業を開始する前に、`geo maintenance mode` フラグの値をtrueに設定してください。メンテナンスが完了したら、システムの正常性を確認後、フラグを false にリセットします。

メンテナンスモードフラグの設定方法については、メンテナンスモードの有効化または無効化（545 ページ） を参照してください。

メンテナンスモードが有効になっている場合：

- ポッドモニタリングやラックサーバーからの BFD リンクモニタリングなどの自動 GR スイッチオーバーはサポートされていません。
- ラック（メンテナンスモードが有効の場合）からパートナーラックへの、CLI ベースの GR スイッチオーバーのみがサポートされます。
- パートナーラックからメンテナンスモードが有効になっているラックに対する、CLI ベースを含む GR スイッチオーバーはサポートされていません。
- 両方のパートナーラックがメンテナンスモードの場合、GR スイッチオーバーはサポートされません。
- すべてのモニタリングアクティビティが一時停止します。
- 従属クラスタは、ローカルの障害を検出してもフェールオーバーをトリガーできません。
- レプリケーション アクティビティはクラスタで続行されます。

- メンテナンスモードでは、サイトのインスタンスロールは暗黙的に変更されませんが、ロールの変更は **geo switch-role role** CLI コマンドを使用して可能です。

  メンテナンス中のクラスタとの間で GR トリガーの送受信はサポートされていません。メンテナンス中のクラスタからは、CLI ベースのフェールオーバーのみがサポートされます。

制限事項

この機能には、本リリースにおいて次の制限があります。

メンテナンスモード機能は、マルチコンピューティング障害スイッチオーバーケースを上書きしません。ただし、パートナーラックもメンテナンスモードである場合は、マルチコンピューティング障害スイッチオーバーシナリオがサポートされます。

（注） マルチコンピューティング障害は、ラック内の複数のサーバーに障害が発生し、その結果パートナーラックがトラフィックを処理する場合に行われるスイッチオーバーケースです。

メンテナンスモードの有効化または無効化

この機能を有効または無効にするには、次のコマンドを実行します。

geo maintenance mode { true | false }

注：

geo maintenance mode { true | false }：メンテナンスモードを有効にするか無効にするかを指定します。メンテナンスモードを有効にするには、true を指定します。メンテナンスモードフラグが true に設定されている場合、クラスタロールが変更され、ラックに対して GR がトリガーされます。

メンテナンスモードの値は、`etcd` ファイルに保存されます。

メンテナンスモードの有効化または無効化の例

次に、コマンドの例を示します。

```
geo maintenance mode true
```

```
geo maintenance mode false
```

メンテナンスモードの状態の確認

メンテナンスモードの状態を確認するには、次の手順を実行します。

```
show geo maintenance mode
result "geo maintenance mode is disabled"
```

アイドル/アクティブの部分的な CDL 更新コールフロー

機能説明

cnSGW-C は、サブスクライバデータの部分的な CDL 更新をサポートしています。部分的な CDL 更新は、CDL ポッドのサブスクライバデータ処理に対する CPU 要件を抑えるのに役立ちます。

サブスクライバのアイドル状態からアクティブ状態への移行およびアクティブ状態からアイドル状態への移行ごとに、cnSGW-C は次のフィールドのみを更新します。

- eNodeB FTEID
- サブスクライバの状態
- ベアラーの状態

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C は、Update Bearer 情報を CDL のフラグデータベースに保存します。

cnSGW-C は、サブスクライバが次のように移行すると部分的な CDL 更新を使用します。

- Release Access Bearer Request の受信時にアクティブ状態からアイドル状態になる
- Modify Bearer Request の受信時にアイドル状態からアクティブ状態になる

部分的な CDL 更新では、session-state-flag には **cdl show sessions summary slice-name <n>** CLI 出力の次の値が表示されます。

- sgw_active：セッションがアクティブな場合
- sgw_inactive：セッションがアイドルの場合

次に、アクティブセッションの出力例を示します。

```
cdl show sessions summary slice-name 1
message params: {session-summary cli session {0 100  0 [] 0 0 false 4096 [] []} 1}
session {
    primary-key 2#/#imsi-123456789012348
    unique-keys [ "2#/#16777229" ]
    non-unique-keys [ "2#/#id-index:1:0:32768" "2#/#id-value:16777229"
    "2#/#imsi:imsi-123456789012348" "2#/#msisdn:msisdn-223310101010101"
    "2#/#imei:imei-123456786666660" "2#/#upf:209.165.201.1"
    "2#/#upfEpKey:209.165.201.1:209.165.201.30" "2#/#s5s8Ipv4:209.165.202.129"
"2#/#s11Ipv4:209.165.201.1"
    "2#/#namespace:sgw" ]
   flags [ byte-flag1:00:13:03:53:00:00:06:85:0A:01:01:1B session-state-flag:sgw_active
 ]
    map-id 1
```

```
    instance-id 1
    app-instance-id 1
    version 1
    create-time 2022-01-20 11:37:15.181259564 +0000 UTC
    last-updated-time 2022-01-20 11:37:15.703032336 +0000 UTC
    purge-on-eval false
    next-eval-time 2022-01-27 11:37:26 +0000 UTC
    session-types [ SGW:rat_type:EUTRAN ]
    data-size 925
```

次に、アイドルセッションの出力例を示します。

```
cdl show sessions summary slice-name 1
message params: {session-summary cli session {0 100  0 [] 0 0 false 4096 [] []} 1}
session {
    primary-key 2#/#imsi-123456789012348
    unique-keys [ "2#/#16777229" ]
    non-unique-keys [ "2#/#id-index:1:0:32768" "2#/#id-value:16777229"
    "2#/#imsi:imsi-123456789012348" "2#/#msisdn:msisdn-223310101010101"
    "2#/#imei:imei-123456786666660" "2#/#upf:209.165.201.1"
"2#/#upfEpKey:209.165.201.1:209.165.201.30"
    "2#/#s5s8Ipv4:209.165.202.129" "2#/#s11Ipv4:209.165.201.1" "2#/#namespace:sgw" ]
    flags [ byte-flag1:00:25:00:55:00:65 session-state-flag:sgw_inactive ]
    map-id 1
    instance-id 1
    app-instance-id 1
    version 3
    create-time 2022-01-20 11:37:15.181259564 +0000 UTC
    last-updated-time 2022-01-20 11:37:18.102852792 +0000 UTC
    purge-on-eval false
    next-eval-time 2022-01-27 11:37:28 +0000 UTC
    session-types [ SGW:rat_type:EUTRAN ]
    data-size 1644
```

制限事項

cnSGW-C は、IPv6 TEID の部分的な CDL 更新をサポートしていません。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
  cdl
    datastore datastore_session_name
      slot metrics report-idle-session-type { true | false }
      end
```

注：

- **slot metrics report-idle-session-type { true | false }**：CDL db_records_total のアイドルセッション数またはアクティブセッション数を有効または無効にします。

設定例

次に、設定例を示します。

```
config
   cdl
      datastore session
         slot metrics report-idle-session-type true
         end
```

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

部分的CDLアップデートアイドルアクティブコールフロー機能では、次の統計情報がサポートされています。

Grafana クエリを実行して、cnSGW-C のアイドルとアクティブなセッションカウントを見つけます。

```
SGW ACTIVE COUNT :-
(db_records_total{namespace=~"$namespace",session_type=~"sgw_active"}) SGW_IDLE
 COUNT : (db_records_total{namespace=~"$namespace",session_type=~"sgw_inactive"})
```

DLDR スロットリングのサポートを含む PFCP セッションレポート

機能説明

ライブネットワーク展開では、一部の外部イベントにより、ほとんどすべてのアイドルセッションが同時にアクティブになります。アイドルセッションがアクティブになると、UPFは、レポートタイプ DLDR のセッションレポート要求を cnSGW-C に送信します。

cnSGW-C がレポートタイプを DLDR とするセッションレポートを受信すると、cnSGW-C がUEのページングのためのDDNメッセージを送信します。UEをアクティブにするため、MMEがUEのページング手順を開始します。ページングが成功した場合、MMEはサービス要求「ベアラー変更要求」を開始します。UE にデータが送信されると、UE はアクセスベアラーのリリース要求を開始し、アイドル状態になります。このコールフローは、システム全体の負荷を増加させます。コールフロー全体が短時間のうちにすべてのサブスクライバで発生すると、システム上で大きなプロセスオーバーヘッドが発生します。

DLDRスロットリングのサポート機能を備えたPFCPセッションレポートにより、cnSGW-Cはシステムに到着するセッションレポート要求の数を制限して、プロセスの過負荷を防ぐことができます。

重要 この機能により、緊急通報もスロットルされます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C は、SBA 過負荷制御に app-infra 機能を使用して、着信メッセージをスロットリングします。

システムは、インターフェイスレベルのしきい値設定に基づいて着信メッセージに適切なアクションを実行します。着信メッセージはそれぞれ異なるキューに追加され、メッセージレベルの優先順位設定に基づいて処理されます。

過負荷サポートの詳細については、UCC 5G セッション管理機能の設定および管理ガイド [英語] の「SMF Overload Support」の章を参照してください。

緊急コール、音声コール、および空のコールに関するセッションレポートをスロットリングから除外できます。

これらのセッションレポートを除外するには、profile sgw. で ddn delay-exclude-arplist の設定を設定します。設定済みのいずれかの ARP に関するセッションレポートを受信すると、cnSGW-C はそのセッションレポートをセッション レポート スロットリングから除外します。

機能設定

この機能を設定するには、次の構成を使用します。

```
config
   instance instance-id instance_id
      endpoint pfcp
         interface sxa
            overload-control msg-type session-report
            rate-limit rate_limit
            queue-size queue_size
            reject-threshold reject_threshold
            pending-request pending_request
            exit
config
   profile sgw sgw_name
      ddn delay-exclude-arplist number_priorities
      end
```

注：

- **overload-control msg-type session-report**：インターフェイスの過負荷に対する仮想メッセージ仕様を設定します。
- **rate-limit** *rate_limit*：仮想キューのレート制限を指定します。
- **queue-size** *queue_size*：各仮想キューのパケット数または容量を指定します。
- **reject-threshold** *reject_threshold*：保留中のリクエストがこのしきい値の割合に達した場合に、受信メッセージを拒否するための上限を指定します。
- **pending-request** *pending_request*：仮想キュー内の保留中リクエスト数を指定します。

- **ddn delay-exclude-arplist** *number_priorities*：DDN/セッション レポート スロットリングの遅延から除外する必要がある、割り当ておよび保持の優先順位のレベルを指定します（1 ～ 15）。

重要

- 両方の GR インスタンスに同じレート制限を設定する必要があります。
- セッション レポート スロットリングは、キュー内の待機時間がトランザクションSLA（保留要求/レート制限 < Txn SLA（秒単位））を超えないようにする必要があります。
- この設定を追加すると、プロトコルポッドが再起動されます。

設定例

次に、設定例を示します。

```
config
   instance instance-id 1
      endpoint pfcp
         interface sxa
            overload-control msg-type session-report
            rate-limit 4500 queue-size 2500 reject-threshold 80 pending-request 2400
            exit
         profile sgw sgw1
         ddn delay-exclude-arplist [ 9 ]
         end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
#show running-config instance instance-id 1 endpoint pfcp
instance instance-id 1
endpoint pfcp
.
.
.
interface sxa
.
.
.
overload-control msg-type session-report
msg-priority high rate-limit 5 priority 1 queue-size 11 reject-threshold 80 pending-request
 10
exit
.
.
.
```

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

DLDR スロットリングサポート機能を備えた PFCP セッションレポートでは、次の統計がサポートされています。

プロトコルのポッドレベルの SXa 統計

```
クエリ：sum(irate(proto_udp_req_msg_total{interface_type="SXA",
message_name=~"session_report_.*"}[1m])) by (message_name, status)
```

```
凡例：{{message_name}} | {{status}}
```

- **accepted**：exclude-arp リストで ARP が設定されているため、受け入れられたセッションレポート
- **throttle_allow**：レート制限フレームワークによって許可されるセッションレポート
- **throttled_pending_req_limit**：レート制限フレームワークによってスロットリングされたセッションレポート。

SGW サービスレベル セッション レポートの統計

```
クエリ：sum(irate(sgw_sx_session_report_stats{}[30s])) by (sx_session_report_type,
 status)
```

```
凡例：{{sx_session_report_type}} -> {{status}}
```

ワンポイント アドバイス

SMF のバルク統計情報のサポートの詳細については、UCC 5G SMF メトリックリファレンス [英語] を参照してください。

cnSGW-C のバルク統計情報のサポートの詳細については、UCC 5G cnSGW-C メトリックリファレンス [英語] を参照してください。

S11 インターフェイスでのセッション作成要求に対するスロットリングのサポート

機能説明

ライブネットワーク展開では、トラフィックの多いモバイルネットワークが原因で、S11 GTPC エンドポイントは、特にピーク時や人が密集するエリアで、多数のセッション作成要求メッセージを受信する可能性があります。

GTPC は、S11 インターフェイスの GTPC ポッドで多数の CSReq を受信すると、すべての要求を受け入れて処理しようとします。そのため、システム全体の負荷が増加する可能性があります。複数の CSReqs コールが短時間に発生すると、システムに大きなプロセスオーバーヘッドが発生します。S11 GTPC エンドポイントのレート制限構成により、GTPC はシステムに入る CSReq の数を制限して、システムのメモリ不足の状態を防ぐことができます。

機能の仕組み

レート制限設定を使用すると、GTPC エンドポイントへの着信 Create Session Request メッセージのフローを制御できるため、GTPC エンドポイントで大量のトラフィックを効果的に処理できます。

GTPC ポッドは、新しいトランザクションの作成中に仮想 ID を追加する app-infra 機能を使用し、インターフェイスと要求のタイプに基づいて着信メッセージをスロットリングします。

システムは、インターフェイスレベルのしきい値設定に基づいて着信メッセージに適切なアクションを実行します。着信メッセージはさまざまなキューに追加され、設定に基づいて処理されます。

設定されたしきい値に達すると、CSReq が拒否され始めます。この設定は、GTPC エンドポイントで常に処理できる Create Session Request の数を制御することで、大量のトラフィックを管理するのに役立ちます。

セッション作成要求に対するスロットリングの有効化

レート制限設定を使用すると、S11 インターフェイスを介して GTPC エンドポイントへの着信セッション作成要求メッセージのフローを管理できるため、GTPC エンドポイントで大量のトラフィックを効率的に処理できます。

重要 レート制限設定が有効になっている場合、緊急コールと WPS セッションにも適用され、これらのコールのセッション作成要求メッセージが拒否またはドロップされる可能性があります。

注：

- 両方の GR インスタンスに同じレート制限を設定する必要があります。
- セッション作成要求スロットリングは、キュー内の待機時間がトランザクション SLA（保留要求/レート制限 < Txn SLA（秒単位））を超えないようにする必要があります。
  - cnSGW を使用しないコンバージドコア展開の場合は、S5 インターフェイスにのみセッション作成要求スロットリング設定を追加することを推奨します。
  - cnSGW を使用するコンバージドコア展開の場合は、S11 および S5 インターフェイスに同時にセッション作成要求スロットリング設定を追加することを推奨します。
- この設定を追加すると、gtpc-ep ポッドが再起動されます。

GTPC エンドポイントでセッション作成要求メッセージのレート制限を有効にするには、次の手順に従ってください。

手順

ステップ 1 インスタンス識別子を指定します。

instance instance-id *instance_id*

例：

```
[smf] smf# config
[smf] smf(config)# instance instance-id 1
```

ステップ 2 GTP エンドポイントを設定します。

endpoint gtp

ステップ 3 インターフェイスを指定します。

interface s11

ステップ 4 セッション作成要求メッセージの過負荷制御を有効化します。

overload-control msg-type create-session-request

ステップ 5 仮想キューの要求がしきい値に達した場合に、セッション作成要求メッセージを拒否するアクションを指定します。

reject-action reject_req

ステップ 6 必要に応じて、仮想キューのレート制限およびその他のパラメータを設定します。

rate-limit *rate_limit* **queue-size** *queue_size* **reject-threshold** *reject_threshold* **pending-request** *pending_requests*

例：

```
[smf] smf(config-interface-s5)# rate-limit 3000 queue-size 8000 reject-threshold 100
pending-request 5000
```

設定例

次に、設定例を示します。

```
 config
   instance instance-id 1
      endpoint gtp
         interface s11
            overload-control create-session-request
            reject-action reject-req
            rate-limit 3 queue-size 10 reject-threshold 100 pending-request 2
            exit
        exit
    exit
exit
```

設定の確認

構成を確認するには、次の show コマンドを使用します。

```
#show running-config instance instance-id 1 endpoint gtp
instance instance-id 1
endpoint gtp
.
.
.
interface s11
.
.
.
overload-control msg-type create-session-request
reject-action reject-req
rate-limit 3 queue-size 10 reject-threshold 100 pending-request 2
exit
.
.
.
```

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

作成セッション要求スロットリング機能では、次の統計情報がサポートされています。

- **rej_req_throttled_pending_req_limit**：保留中の要求の制限に達したときに拒否された作成セッション要求メッセージの総数。
- **rej_req_throttled_queue_full**：トランザクションキューがいっぱいのときに拒否された作成セッション要求メッセージの総数。

次に、cnSGW サービスレベルメトリックの例を示します。

```
virtual_message_Pending_req_total{app_name="SGW",cluster="Local",data_c
enter="DC",instance_id="0",interface="s11",msg_type="createsessionrequest",
service_name="s11-gtpc-ep1",virtual_msg_id="1"} 0
```

```
virtual_message_dequeue_rate_total{app_name="SGW",cluster="Local",data_c
enter="DC",instance_id="0",interface="s11",msg_type="createsessionrequest",
service_name="s11-gtpc-ep1",virtual_msg_id="1"} 11
virtual_message_drop_total{app_name="SGW",cause="PendingRequestsLimitReached",cluster="Local",data_center="DC",instance_id="0",interface="s11",
msg_type="createsessionrequest",service_name="s11-gtpcep1",
virtual_msg_drop_code="8004",virtual_msg_id="1"} 19
<<<<<<<virtual_message_queued_total{app_name="SGW",cluster="Local",data_cent
er="DC",instance_id="0",interface="s11",msg_type="createsessionrequest",ser
vice_name="s11-gtpc-ep1",virtual_msg_id="1"} 11
<<<<<virtual_message_rate_limit_reached_total{app_name="SGW",cluster="Local",d
ata_center="DC",instance_id="0",interface="s11",msg_type="createsessionreq
uest",service_name="s11-gtpc-ep1",virtual_msg_id="1"} 6
```

復元力処理

機能説明

復元力処理機能により、CLI 制御のフレームワークが導入され、システム障害を確認した場合、またはクラッシュが報告された場合に、サービスポッドのリカバリがサポートされるため、次のいずれかのサービスポッドのリカバリに役立ちます。

- sgw-service ポッド
- smf-service ポッド
- gtpc-ep ポッド
- プロトコルポッド

これらのサービスポッドは、複数のセッションメッセージを処理するロジックを含むソフトウェアモジュールです。サービスポッドは、次のいずれかのシナリオ、または複数のシナリオの組み合わせが原因で障害が発生しやすくなります。

- 複雑なコールフローおよびコリジョン処理
- セッション状態が不整合性
- セッション状態に対する着信メッセージの不適切な処理
- 受信メッセージ内の予期しないコンテンツや未処理のコンテンツ

システム障害やクラッシュを確認するたびに、障害の動作によってサービスポッドが強制的に再起動され、他のセッションの進行中のトランザクション処理に影響を及ぼします。ポッドの再起動後でもクラッシュは再発します。

このリスクを軽減するには、アクションが定義されている CLI ベースのフレームワークを使用し、サブスクライバセッションをクリーンアップするか、現在の処理を終了します。

機能の仕組み

ここでは、障害リカバリフレームワークを使用してクラッシュに対するアクションを定義する方法について説明します。このフレームワークでは、次のアクションのいずれかを定義できます。

- [Terminate]：障害が発生した場合、このアクションは障害が発生したトランザクションを終了し、サブスクライバセッションのキャッシュをクリアします。これは、smf-service ポッドおよび sgw-service ポッドに適用されます。

（注） ポッドは再起動されません。このアクション中にデータベースはクリアされません。

- [Cleanup]：障害が発生した場合、このアクションは障害が発生しているサブスクライバセッションをクリアし、コールをリリースします。これは、smf-service ポッドおよび sgw-service ポッドに適用されます。
- [Graceful reload]：障害が発生した場合、このアクションはポッドを再起動します。これは、gtpc-ep ポッド、protocol ポッド、および ポッドに適用されます。障害信号を処理して、キープアライブポートなどのリソースをクリーンアップし、初期段階で閉じます。また、checkport スクリプトがポッドの状態を検出し、対応するポッドの VIP スイッチ処理を開始できるようにします。
- [Reload]：ポッドがクラッシュすると、リロードアクティビティが開始されます。これは、すべてのポッドに適用されるデフォルトの設定または値です。

機能設定

この機能を設定し、システムの障害回復を有効にするには、次の設定例を使用します。

```
config
   system-diagnostics {   gtp | pfcp | service | sgw-service }
      fault
         action { abort | cleanup { file-detail | interval | num |
skip { ims | emergency | wps } } | graceful-Reload | reload }
         end
```

注：

- **system-diagnostics { gtp | pfcp | service | sgw-service }**：システム診断に必要なサービスポッドのタイプを指定します。使用可能なポッドオプションは、、gtp、pfcp、smf-service、および sgw-service です。
- **fault**：セッションの処理中に障害回復を有効にします。
- **action { abort | cleanup | graceful-Reload | reload }**：障害発生時に実行するアクションを次から 1 つ指定します。デフォルトアクションはリロードです。

- **abort**：障害のあるトランザクションを削除し、そのセッションキャッシュをクリアします。データベースはクリアされません。

（注） これは、smf-service ポッドに対する排他的オプションです。

- **cleanup { file-detail | interval | num | skip }**：クリーンアップ アクティビティを有効にします。障害を軽減するためのアクションとして次の選択肢があります。
  - **file-detail**：ファイル名に行番号を付けて一覧表示します。リカバリからファイル名の詳細が除外されます。
  - **interval**：間隔を分単位で指定します。この期間に、障害の最大数を許可する許容間隔を指定します。1 ～ 3600 の範囲の整数で指定する必要があります。
  - **num**：1 つの間隔内で許容可能な障害の最大数を指定します。0 ～ 50 の範囲の整数で指定する必要があります。
  - **skip { ims | emergency | wps }**：アクティブ音声コール、WPS、または緊急コールに対するサブスクライバセッションのクリーンアップのスキップを有効にします。
    - アクティブ音声コールを検出するには、次のコマンドを使用します。

      profile dnn *dnn_name* **ims mark qci** *qos_class_id*

    - クリーンアップのスキップ設定を有効にすると、SMF は問題のあるトランザクションを削除し、そのセッションキャッシュをクリアします。
    - セッションのセットアップ中またはリリース状態の間に障害が発生すると、SMF は次の処理を実行します。

      ：セッション終了時にトランザクションを削除します。

      ：これらの状態の間、設定済みの障害アクションをオーバーライドします。

      ：障害のあるトランザクションのセッションキャッシュおよびデータベースエントリをクリアします。

    - これにより、動的な設定変更が可能になります。

（注） これは、smf-service および sgw-service ポッドの排他的オプションです。

- **graceful-Reload**：ポッドの正常なリロードのためのオプションを指定します。サービスポッドは、障害信号を処理してキープアライブポートなどのリソースをクリーンアップし、クラッシュ処理（ポッド再起動処理）を続行します。

（注） これは、、gtpc-ep、および protocol サービスポッドに対する排他的なオプションです。

- **reload**：障害動作が原因でポッドがクラッシュした場合に、ポッドをリロードします。これは、すべてのサービスポッドに適用できるオプションです。これはデフォルトのオプションでもあります。

設定例

次の設定例では、障害発生アクションをサブスクライバのクリーンアップとして、smf-service または sgw-service ポッドのクラッシュが 10 分間隔以内で 3 回許可されます。

```
config
    system-diagnostics { service | sgw-service }
        fault
            num 3 interval 10
            action cleanup
            end
```

次の設定例では、 またはgtpc-epポッド、あるいはprotocolポッドのグレースフル障害処理で、障害信号の受信時にキープアライブポートを閉じることができます。

設定の確認

設定を確認するには、次のコマンドを実行します。

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

復元力処理機能では、次のバルク統計がサポートされています。

recover_request_total：この統計には、次の新しいラベルが含まれています。

- **action**：障害アクションを定義します。
- **reason**：障害の理由を定義します。
- **status**：障害ステータスを定義します。

次に、復元力処理機能のバルク統計の例を示します。

```
recover_request_total{action="panic_recovery_cleanup",
app_name="SMF",cluster="Local",data_center="DC",instance_id="0", reason="creating
 panic",service_name="sgw-service",status="success"} 1
```

SMF のバルク統計のサポートの詳細については、UCC 5G SMF メトリックリファレンス [英語] を参照してください。

cnSGW-C のバルク統計のサポートの詳細については、UCC 5G cnSGW-C メトリックリファレンス [英語] を参照してください。

ローミングピアパス管理の最適化

機能説明

cnSGW-C は着信ローミングをサポートします。着信ローミングが有効になっている場合、cnSGW-C はローミングのホームネットワークにあるリモート PGW と通信します。

cnSGW-C はローミングピアに向けてエコー要求メッセージを生成し、パス障害を検出することで、ローミングピアからのエコー要求メッセージを処理します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C は、サブスクライバポリシーとオペレータポリシーを使用して、ピアをローミングピアまたはホームピアとして分類します。cnSGW-C は、ローミングピアに次の機能を適用します。

- cnSGW-C は、ローミングピアから受信したエコー要求メッセージにすぐに応答します。エコー要求で再起動カウンタ値が変更された場合、cnSGW-C はピアへのパス障害を検出しません。
- cnSGW-C は、設定されたエコー間隔に達した後も、ローミングピアに対してエコー要求を生成し続けます。エコー応答で再起動カウンタ値が変更された場合、cnSGW-C はピアへのパス障害を検出します。
- 最初のセッション作成応答メッセージおよび SGW 再配置ベアラー変更応答メッセージで再起動カウンタ値が変更された場合、cnSGW-C はピアへのパス障害を検出します。
- cnSGW-C は、ローミングピアからエコー要求を受信しても、ローミングの最後のアクティビティ時間を更新しません。
- 以前の実装では、Nodemgr の環境変数 ROAMING_PEER_ECHO_MODULATOR がローミングピアに対するエコー要求の生成を制御します。この環境変数の値を変更すると、変更後に毎回 NodeMgr ポッドが再起動します。

  この動作を最適化するために、環境変数 ROAMING_PEER_ECHO_MODULATOR の使用は廃止され、S8e インターフェイス設定を使用してローミングピアに異なるエコー間隔を設定するためのサポートが追加されます。S8eインターフェイスのエコー設定が有効になっている場合、cnSGW-C はこの設定を使用してローミングピアに対するエコー要求を生成します。

詳細については、「ローミングピアノードのエコー要求最適化の有効化」を参照してください。

- S8e インターフェイス設定が存在しない場合、cnSGW-c は環境変数 ROAMING_PEER_ECHO_MODULATOR を使用して、ローミングピアに対するエコー要求の生成を制御します。ROAMING_PEER_ECHO_MODULATOR のデフォルト値は 3 です。cnSGW-C は、 ROAMING_PEER_ECHO_MODULATOR * エコー間隔に到達した後、ローミングピアに対してエコー要求を生成します。

  たとえば、ROAMING_PEER_ECHO_MODULATOR が 3 で、エコー間隔が 60 の場合、cnSGW-C は 180 秒後にエコー要求を生成します。同様に、ROAMING_PEER_ECHO_MODULATOR が 0 の場合、cnSGW-C はローミングピアに対してエコー要求を生成しません。

- GTPC-EP ポッドでは、変数 GTPC_UPDATE_LAST_MSG_RECV_TIME_AFTER が最後のアクティビティ時間の更新を制御します。この変数のデフォルト値は 30 秒です。cnSGW-C は、30 秒後にピアの最後のアクティビティ時間を更新します。この値を大きくすると、NodeMgr への最後のアクティビティ時間の更新通知を減らすことができます。

機能設定

この機能の設定には、次の手順が含まれます。

- ローミングステータスをローミングするオペレータポリシーを構成し、オペレータポリシーをサブスクライバポリシーに関連付けて、オペレータをローミングピアとして識別します。詳細については、オペレータポリシーおよびサブスクライバポリシーの構成 （560 ページ）を参照してください。
- BGP ルートをダイナミックに追加する際に、使用するデフォルトゲートウェイを構成します。詳細については、デフォルト ゲートウェイの設定 （562 ページ）を参照してください。
- ローミングピアノードのエコー要求最適化を有効にします。「ローミング ピアノードのエコー間隔の構成」を参照してください。

オペレータポリシーおよびサブスクライバポリシーの構成

この機能を設定するには、次の構成を使用します。

```
config
  policy
    subscriber subscriber_name
      precedence precedence_value
        imsi mcc mcc_value
        imsi mnc mnc_value
        operator-policy policy_name
        exit
      exit
    operator operator_policy
```

```
    roaming-status roamer
    exit
profile sgw sgw_profile_name
    subscriber-policy policy_name
    end
```

注：

- **precedence** *precedence_value*：エントリの優先順位を指定します。1 ～ 2048 の範囲の整数で指定する必要があります。
- **mcc** *mcc_value*：モバイル国コード（MCC）を指定します。3 桁の数字で指定する必要があります。
- **mnc** *mnc_value*：モバイルネットワークコード（MNC）を指定します。2 桁または 3 桁の数字で指定する必要があります。
- **operator-policy** *policy_name*：オペレータポリシー名を指定します。文字列で指定する必要があります。
- **policy-operator** *operator_policy*：オペレータポリシーを指定します。以下のいずれかを指定する必要があります。
  - <any string>
  - defOprPoll
  - opPolHomer
  - opPolRoaming
  - opPolVisiting
  - opPolVisiting_hrt
  - opPolVisiting_hrt_overriden
- **roaming-status roamer**：ピアのローミングステータスを指定します。デフォルトでは、無効になっています。
- **subscriber-policy** *policy_name*：サブスクライバポリシー名を指定します。文字列で指定する必要があります。

設定例

次に、設定例を示します。

```
config
   policy subscriber polSubSgw
      precedence 1
         imsi mcc 310
         imsi mnc 260
         operator-policy Home_op1
         exit

      precedence 2
         imsi mcc 311
```

```
      imsi mnc 660
      operator-policy Home_op1
      exit

   precedence 3
      imsi mcc 310
      imsi mnc 240
      operator-policy Home_op1
      exit

   precedence 4
      operator-policy Roaming_SGW_op1
      exit
   exit

   policy operator Roaming_SGW_op1
      roaming-status roamer
      exit

   profile sgw sgw1
      subscriber-policy polSubSgw
      end
```

デフォルト ゲートウェイの設定

この機能を設定するには、次の構成を使用します。

```
config
   router bgp local_as_number
   policy-name policy_name
      source-prefix source_prefix_value
      mask-range mask_range
      interface interface_id
      gateWay gateway_address
      end
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **policy-name** *policy_name*：ポリシー名を指定します。
- **source_prefix** *source_prefix_value*：送信元プレフィックスの値を指定します。
- **mask-range***mask_range*：マスクの範囲を指定します。
- **gateWay** *gateway_address*：ゲートウェイアドレスを指定します。

設定例

次に、設定例を示します。

```
config
   router bgp 6500
   policy-name sgw_bgp
      source-prefix 209.165.201.12/32
      mask-range 32..32
      interface ens224.2084
```

```
gateWay 209.165.201.28
end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

- **show subscriber namespace sgw imsi 123456789012345 full subscriber-details**

```
{
"subResponses": [
  "pdnInfoList": {
     "pdnInfo": [
          {
           "plmnType": "VISITOR"
           "s5ePeerType": "ROAMER"
           }
       ]
   }
]
}
```

- **show peers | include S5**

```
1 S5E 209.165.201.12:212320.20.20.124:2123Inbound nodemgr-1 Udp 2 minutes PGW
MaxRemoteRcChange: N/A,Recovery: 10 S5
1 S5E 209.165.201.12:212320.20.20.127:2123Inbound nodemgr-0 Udp 35 seconds PGW
MaxRemoteRcChange: N/A,Recovery: 10,S5E PeerType: Roaming
```

ローミングピアノードのエコー要求最適化の有効化

ローミングピアノードのエコー要求の最適化を有効にするには、次の手順を実行します。

手順

ステップ 1 S8e インターフェイスでエコー間隔を設定します。
ステップ 2 パス障害検出ポリシーを設定します。
ステップ 3 S8e インターフェイスでのパス障害検出ポリシーの関連付け （565 ページ）。

S8e インターフェイスのエコー間隔の設定

S8e インターフェイスでエコー要求を有効にするには、次の手順を使用します。

手順

ステップ 1 インスタンス ID コンフィギュレーション モードを開始します。

instance instance-id *instance_id*

例：

```
[smf] smf# config
[smf] smf(config)# instance instance-id 1
```

ステップ 2 GTPC エンドポイントを設定します。

endpoint *endpoint_type*

例：

```
[smf] smf(config-instance-1)# endpoint gtp
```

ステップ 3 S8e インターフェイスを設定します。

interface *interface_type*

例：

```
[smf] smf(config-endpoint-gtp)# interface s8e
```

ステップ 4 エコーを設定します。

echo interval *echo_interval* **retransmission-timeout** *retransmission_timeout* **max-retransmissions** *max_retransmissions*

例：

```
[smf] smf(config-interface-s8e)# echo interval 300 retransmission-timeout 10
max-retransmissions 4
```

（注）

- *echo_interval*：エコー間隔を秒単位で指定します。60 ー 3600 の範囲の整数で指定する必要があります。デフォルト値は 300 です。ローミングピアへのエコー要求の生成を無効にするには、0 を使用します。
- *retransmission_timeout*：エコー再送信のタイムアウトを秒単位で指定します。1 ～ 20 の範囲の整数で指定する必要があります。デフォルト値は 5 です。
- *max_retransmissions*：GTP エコー要求の最大再試行回数を指定します。0 ～ 15 の範囲の整数で指定する必要があります。デフォルト値は 3 です。

ステップ 5 設定を保存してコミットします。

例：

```
[smf] smf(config-interface-s8e)# end
```

ステップ 6 （任意）**show running-config instance instance-id 1 endpoint gtp interface s8e** コマンドを使用して、エコー間隔が設定されているか確認します。

例：

```
[sgw] smf# show running-config instance instance-id 1 endpoint gtp interface s8e
Thu Dec  12 09:26:46.295 UTC+00:00
instance instance-id 1
 endpoint gtp
  interface s8e
   echo retransmission-timeout 10
   echo max-retransmissions 4
   echo interval          300
```

```
  exit
 exit
exit
```

S8e インターフェイスでのパス障害検出ポリシーの関連付け

S8e インターフェイスでエコー要求を有効にするには、次の手順を使用します。

手順

ステップ 1 インスタンス ID コンフィギュレーション モードを開始します。

instance instance-id *instance_id*

例：

```
[smf] smf# config
[smf] smf(config)# instance instance-id 1
```

ステップ 2 GTP エンドポイントを設定します。

endpoint *endpoint_type*

例：

```
[smf] smf(config-instance-1)# endpoint gtp
```

ステップ 3 s8e インターフェイスを設定します。

interface *interface_type*

例：

```
[smf] smf(config-endpoint-gtp)# interface s8e
```

ステップ 4 パス障害検出ポリシーを定義します。

path-failure detection-policy *detection_policy_name*

例：

```
[smf] smf(config-interface-s8e)# path-failure detection-policy pf1
```

（注）
detection_policy_name：障害検出ポリシー名を指定します。文字列で指定する必要があります。

ステップ 5 設定を保存してコミットします。

例：

```
[smf] smf(config-interface-s8e)# end
```

ステップ 6 （任意）**show running-config instance instance-id 1 endpoint gtp interface s8e** コマンドを使用して、パス障害検出ポリシーが有効になっているか確認します。

例：

```
[sgw] smf# show running-config instance instance-id 1 endpoint gtp interface s8e
Thu Dec  12 09:26:46.295 UTC+00:00
instance instance-id 1
 endpoint gtp
  interface s8e
   echo retransmission-timeout 10
   echo max-retransmissions 4
   echo interval          300
   path-failure detection-policy pf1
  exit
 exit
exit
```

パス障害検出ポリシーの設定

パス障害検出ポリシーを設定するには、次の手順を実行します。

手順

ステップ 1 パス障害検出コンフィギュレーション モードを開始します。

path-failure detection-policy*detection_policy_name*

例：

```
[smf] smf# config
[smf] smf(config)# path-failure detection-policy pf1
```

ステップ 2 max-remote-rc-change および ignore を指定して、パス障害検出ポリシーを設定します。

max-remote-rc-change *max_remote_rc_change* **ignore** *ignore*

（注）
ignore：再起動カウンタ値、エコータイムアウト、またはエコー障害を無視するように指定します。**control-rc-change**、**echo-failure**、または **echo-rc-change** にする必要があります。

例：

```
[smf] smf(config-path-failure-detection-pf1)# max-remote-rc-change  2 ignore echo-failure
```

ステップ 3 設定を保存してコミットします。

例：

```
[smf] smf(config-path-failure-detection-pf1)# end
```

ステップ 4 （任意）**show running-config policy path-failure-detection** コマンドを使用して、パス障害検出ポリシーが設定されているか確認します。

例：

```
[sgw] smf# show running-config policy path-failure-detection
Thu Dec  12 09:26:46.295 UTC+00:00
Thu Dec  12 09:27:17.678 UTC+00:00
```

```
policy path-failure-detection pf1
 max-remote-rc-change 2
 ignore echo-failure
exit
```

ローミングピアノード向けエコーの最適化の設定例

ここでは、S8e インターフェイスでエコー最適化を有効にする設定例を示します。

```
config
instance instance-id 1
 endpoint gtp
  interface s8e
   echo retransmission-timeout 10
   echo max-retransmissions 4
   echo interval          300
   path-failure detection-policy pf1
  exit
 exit
exit

config
policy path-failure-detection pf1
 max-remote-rc-change 2
 ignore echo-failure
exit
```

OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

ローミングピアパス管理の最適化機能では、次の統計情報がサポートされています。

ローミングピアからのエコー要求処理が抑制されていることを示す gtpc-ep 統計。

```
gtpc_roaming_peer_path_mgmt{app_name="SGW",cluster="Local",data_center="DC",
gtpc_peer_type="ROAMER",instance_id="1",interface_type="S5E",service_name="gtpc-ep",
 status="suppressed"} 1
```

bgp 追加要求の総数を示す upd_proxy 統計情報。

```
# HELP upd_proxy_bgp_routes_count Total number of bgp add request # TYPE
upd_proxy_bgp_routes_count counter
upd_proxy_bgp_routes_count{app_name="SGW",cluster="Local",data_center="DC",
gr_instance_id="1",instance_id="0",service_name="udp-proxy",status="success"}
 1
```

DB データベース更新にフラグを設定する

機能説明

cnSGW-C、サブスクライバの状態がアイドルからアクティブに変わり、URI、UeTz、UCI、またはサービングネットワークが変更されるたびに、CDL を更新します。

CDL に向けられたトランザクション要求が増加すると、cnSGW-C による CPU 使用率が高くなります。不要な CPU 使用率を防ぐために、cnSGW-C は 変更された属性で CDL のサブセットのみを更新します。

DDN 手順用のフラグ DB データベース

DDN 手順が完了すると、sgw-service は CPU 使用率に影響を与える CDL を更新します。CPU 使用率を最適化するために、部分的な更新でのみ DDN が CDL に通知されます。

DDN 内部タイマー

cnSGW-C は、CDL のタイマー機能を適用することによって、DDN 再試行タイマーを実装します。すべての DDN トランザクションで、完全な CDL インスタンスを更新する必要がある DDN 再試行タイマーが開始されるため、CDL および sgw-service の CPU 使用率が増加します。

cnSGW-C は、sgw-profile から構成可能な統合 DDN 再試行タイマーを搭載するように変更されました。このアプローチにより、cnSGW-C は内部タイマーであるため、DDN 再試行タイマーを開始するために CDL と通信しないため、パフォーマンスが向上します。DDN 再試行タイマーが 10 秒間開始されます。

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスサポートについて説明します。

バルク統計サポート

フラグ DB データベース更新機能では、次の統計がサポートされています。

- mbr_partial_cdl_update：Modify Bearer Request によって呼び出された部分的な CDL 更新手順の合計数をキャプチャします。

  サンプル クエリ：

```
sgw_cdl_update_stats{app_name="smf",cdl_update_type="
mbr_partial_cdl_update",cluster="Local",data_center="DC",gr_instance_id="1",
 instance_id="0",rat_type="EUTRAN",service_name="sgw-service"} 1
```

- ddn_partial_cdl_update：DDN が呼び出した部分的な CDL 更新手順の合計数をキャプチャします。

サンプル クエリ：

```
sgw_cdl_update_stats{app_name="smf",cdl_update_type="
ddn_partial_cdl_update",cluster="Local",data_center="DC",gr_instance_id="1",
 instance_id="0",rat_type="EUTRAN",service_name="sgw-service"} 1
```

バルク統計のサポートの詳細については、UCC サービング ゲートウェイ コントロールプレーン機能メトリックリファレンス [英語] を参照してください。

プロトコルマイクロサービスに統合されたUDPプロキシ機能

機能説明

UDP プロキシマイクロサービスは、UDP プロトコルをトランスポート層プロトコルとして、必要とするプロトコル（PFCP、GTPC、および RADIUS）の UDP トランスポート終端を提供します。UDP プロキシは、ユーザー空間パケット転送と IPC 通信をプロトコルマイクロサービスに提供します。また、送信元IPアドレスのオブザーバビリティのためにホストネットワーキングを使用し、アクティブ/スタンバイモードで動作します。

複数のプロトコルマイクロサービスが、UDP トランスポートの UDP プロキシに依存しているため、UDPプロキシはスケーリングのボトルネックです。メッセージの急増は、パケットのドロップにつながる可能性があります。

着信メッセージと発信メッセージでは、メッセージの転送に UDP プロキシポッドが使用されます。UDP プロキシでは、最小限のパケット処理でメッセージが GTPC-EP ポッドに転送されるため、パケットのマーシャリングまたはマーシャリング解除とともに、メッセージ転送のための IPC 通信が必要です。

スケーリングのボトルネックを軽減するために、UDPプロキシ機能が各プロトコルのマイクロサービスにマージされます。プロトコルポッドはメッセージを直接受信し、メッセージの転送と IPC 通信を回避します。

UDPプロキシバイパスでは、シグナリングパスにおいてマイクロサービス間で1ホップ減らすことで、CPU 使用率を向上させます。cnSGW-C は、PFCP および GTPC プロトコルの UDP プロキシバイパスをサポートしています。

UDP プロキシバイパスを使用した PFCP プロトコルエンドポイント

UDPプロキシモードでは、N4、Sxa、GTP-Uなどのプロトコルに対するメッセージ交換はすべて UDP プロキシを介して実行されます。UDP プロキシは、UPF との接続または接続の受信に関与します。サービスまたは UPF がすべてのノード関連またはセッション関連のメッセージを開始し、応答は UDP プロキシを通過します。UDP プロキシは、すべてのノード関連のメッセージを処理し、各メッセージをプロトコルノードに転送します。

アウトバウンド UDP プロキシバイパスモードでは、セッション関連のメッセージはプロトコルから UPF に直接渡されます。ノード関連のメッセージでは、引き続き現在のパスが使用され、UDP プロキシを経由してプロトコルポッドやノードマネージャに渡されます。

インバウンドおよびアウトバウンドの UDP プロキシバイパスモードでは、サービスは、UDP プロキシをバイパスし、プロトコルポッドを介してセッション関連のメッセージを UPF に直接送信します。プロトコルは、アプリケーションサービスが UPF 宛ての PFCP メッセージを開始する限り、UPF との接続も確立します。

PFCP 用 UDP プロキシバイパスの詳細については、UCC 5G SMF 設定および管理ガイド [英語] を参照してください。

UDP プロキシバイパスを使用した PFCP プロトコルエンドポイント

UDP プロキシモードでは、GTPv2 プロトコルに対するメッセージ交換はすべて UDP プロキシを介して実行されます。UDP プロキシは、S11 と S5e、S2b、および S5/S8 インターフェイスでの接続または受信を行います。サービスまたは GTP ピアがセッション関連またはノード関連のメッセージを開始し、応答は UDP プロキシを通過します。UDP プロキシは、すべてのノード関連のメッセージを処理し、各メッセージをプロトコルノードに転送します。

インバウンドおよびアウトバウンド UDP プロキシバイパスモードでは、サービス開始セッション関連のメッセージは、UDP プロキシをバイパスして、GTPC-EP ポッドを介して GTP ピアに直接送信されます。ノード関連のメッセージの場合、GTPC-EP は、S11、S5e、S2b、および S5/S8 インターフェイス上でピアが接続するために GTP エンドポイントを開始します。GTPC-EP ポッドは、アプリサービスが GTP ピアへの GTPv2 メッセージを開始するときに、として GTP ピアとの接続も確立します。

次の機能は、UDP プロキシから統合されます。

- トランザクション SLA
- GTP パケットの DSCP マーキング
- ローミングサブスクライバの BGP ルートのオンザフライ追加
- ディスパッチャ機能と着信再送信のサポート
- DDN の SGW キャッシュの統合
- MBR キャッシュ統合

次の機能は GTPC-EP から統合されます。

- アウトバウンド要求の n3t3 構成に基づく再送信
- プロトコルのモニタリングとサブスクライバのモニタリング
- エコーメッセージ処理

UDP プロキシモードをサポートする既存の機能はすべて、UDP プロキシバイパスモードの有無にかかわらずサポートされます。

機能の仕組み

ここでは、この機能の仕組みを説明します。

バイパス機能が有効になっている場合、GTPC-EP k8 サービスは無効になります。

UDP プロキシバイパス機能を備えた GTPC プロトコルエンドポイントには、ホスト環境でアクティブ/スタンバイモードの GTPC-EP ポッドが実行されている必要があります。GTPC-EP ポッドがアクティブ-スタンバイモードのホスト環境で実行されている場合、k8-service は無効になります。さらに、ポッド（SGW-Service およびノードマネージャ）が GTPC-EP ポッドと通信する必要がある場合、GTPC-EP ポッドに追加のエンドポイントが必要です。このインフラエンドポイントは GTPC-EP アプリの起動時に初期化され、内部 IP が同じ目的で使用されます。

内部 IP が構成されていない場合、使用可能な GTP VIP がインフラエンドポイントの初期化に使用されます。

S11、S5、S5e、および S2b インターフェイスは、ベース GTP VIP IP アドレスの代わりに GTP VIP を構成するために使用されます。

UDP ソケットは、GTP パケットを処理するために GTPC-EP ポッドで作成されます。

バイパス UDP プロキシ機能を有効または無効にするための新しい CLI またはキーワードは追加されていません。既存のエンドポイント構成を次のように使用して、UDP プロキシのバイパス機能を有効または無効にします。

- GTP VIP は、UDP プロキシ（バイパスなし）を使用するためにエンドポイントプロトコルで構成する必要があります。
- GTP VIP は、バイパス UDP プロキシを有効にするために GTPC エンドポイントで構成する必要があります。
- GTP VIP がプロトコルエンドポイントと GTP エンドポイントの両方で構成されている場合、UDP プロキシがデフォルトで使用されます。
- 再送信 n3t3 ベース、エコー、SLA、ディスパッチャ、および DSCP などの GTPC 機能固有の構成は、バイパス機能に関係なく、エンドポイント GTP で構成する必要があります。

（注） この機能が導入される前は、エンドポイント GTP が構成されている場合、UDP プロキシモードがデフォルトの動作でした。この機能を使用すると、エンドポイント GTP が GTP インターフェイス VIP（s5、s11、s5e、または s2b）で構成されている場合、UDP プロキシバイパスはデフォルトで有効になります。UDP プロキシバイパスを無効にするには、エンドポイントプロトコルを GTP インターフェイス VIP（s5、s11、s5e、または s2b）で構成する必要があります。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

UDP プロキシバイパスを使用した PFCP プロトコルエンドポイントのコールフロー

このセクションでは、UDP プロキシバイパスを使用した PFCP プロトコルエンドポイントのコールフローについて説明します。

図 *107 : UDP* プロキシバイパスを使用した *PFCP* プロトコルエンドポイントのコールフロー

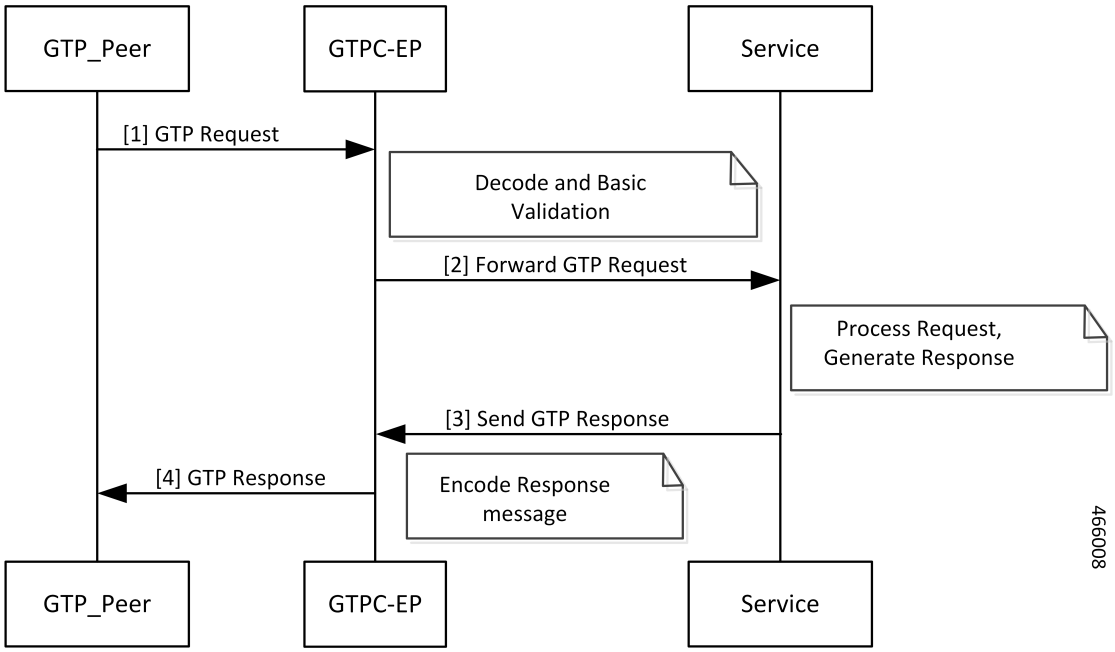


表 *189 : UDP* プロキシバイパスを使用した *PFCP* プロトコルエンドポイントのコールフローの説明

ステップ	説明
1	GTP_Peer が GTPC-EP ポッドに GTP 要求を送信します。
2	GTPC-EP ポッドがメッセージをデコードし、基本的な検証を実行して、GTP 要求をサービスに転送します。
3	サービスが GTP 要求を処理し、GTP 応答を生成して、GTP 応答を GTPC-EP ポッドに送信します。
4	GTPC-EP ポッドが GTP 応答メッセージをエンコードし、GTP 応答メッセージを GTP_Peer に転送します。

機能設定

この機能の設定には、次の手順が含まれます。

- 内部 VIP の構成
- インフラ GTP エンドポイントを初期化するためのインターフェイスでの GTP VIP の構成

内部 *VIP* の構成

内部 VIP を構成するには、次の設定を使用します。

```
config
  instance instance-id instance_id
    endpoint gtp
      internal-vip  vip_ip_address
      vip-ip ipv4_address vip-port ipv4_port_number
      vip-ipv6 ipv6_address vip-ipv6-port ipv6_port_number
      dual-stack-transport { true | false }
      end
```

注：

- **internal-vip** *vip_ip_address*：GTPC-EP ポッドの追加エンドポイントの内部 VIP IP アドレスを指定します。サービスポッドは、この IP アドレスにメッセージを直接送信します。
- **dual-stack-transport { true | false }**：IPv6 または IPv4 アドレスを指定できるデュアルスタック機能を有効にします。この機能を有効にするには、true を指定します。

GTP VIP の設定

インフラ GTP エンドポイントを初期化するためのインターフェイスで GTP VIP を設定するには、次の設定を使用します。

```
config
  instance instance-id instance_id
    endpoint gtp
      vip-ip ipv4_address vip-port ipv4_port_number
      vip-ipv6 ipv6_address vip-ipv6-port ipv6_port_number
      dual-stack-transport { true | false }
    interface interface_name
      vip-ip ipv4_address vip-port ipv4_port_number
      vip-ipv6 ipv6_address vip-ipv6-port ipv6_port_number
      dual-stack-transport { true | false }
      end
```

注：

- **vip-ip** *ipv4_address* **vip-port***ipv4_port_number*：インターフェイスの IPv4 アドレスを指定します。
- **vip-ipv6** *ipv6_address* **vip-ipv6-port***ipv6_port_number*：インターフェイスの IPv6 アドレスを指定します。
- **dual-stack-transport { true | false }**：IPv6 または IPv4 アドレスを指定できるデュアルスタック機能を有効にします。この機能を有効にするには、true を指定します。

設定例

次に、構成例を示します。

UDP プロキシバイパスが有効になっている GTPC-EP の内部 VIP の構成例：

```
config
    instance instance-id 1
        endpoint gtp
          internal-vip 209.165.201.15
          end
```

バイパス機能を有効にした（UDP プロキシがバイパスされた）構成例：

```
config
    instance instance-id 1
        endpoint gtp
          vip-ip 209.165.201.20
          interface s5
            vip-ip 209.165.201.20
            exit
          interface s5e
            vip-ip 209.165.201.8
            exit
          interface s2b
            vip-ip 209.165.201.20
            exit
          interface s11
            vip-ip 209.165.201.8
            end
```

バイパス機能が無効になっている構成例（GTP メッセージに UDP プロキシを使用）：

```
config
    instance instance-id 1
        endpoint protocol
          vip-ip 209.165.201.20
          interface s5
            vip-ip 209.165.201.20
            exit
          interface s5e
            vip-ip 209.165.201.8
            exit
          interface s2b
            vip-ip 209.165.201.20
            exit
          interface s11
            vip-ip 209.165.201.8
            end
```

第 41 章

Presence Reporting Area

- 機能の概要と変更履歴 （575 ページ）
- 機能説明 （575 ページ）
- 機能の仕組み （576 ページ）

機能の概要と変更履歴

要約データ

表 *190* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *191* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.1

機能説明

Presence Reporting Area（PRA）は、ポリシー制御や課金理由でエリア内の UE プレゼンスをレポートするために、3GPP パケットドメイン内で定義されるエリアです。

Presence Reporting Area の構成は次のとおりです。

- E-UTRAN の TA または eNB や ECGI
- UTRAN の RA または SA あるいは CGI
- GERAN の RA または CGI

MME プールに適用される Presence Reporting Area には、UE 専用の Presence Reporting Area とコアネットワークの事前設定された Presence Reporting Area の 2 種類があります。

（注） cnSGW-C は、このリリースでコアネットワークの事前設定された PRA をサポートしています。

PRA を活用して異なる課金ポリシーの適用を実装する方法については、「3GPP TS 23.401」を参照してください。

cnSGW は以下をサポートしています。

- CSRsp、CBReq、UBreq、MBRsp、および CNRsp メッセージで PGW から受信した MME への PRA アクションの受け渡し。
- CSReq、CBRsp、UBRsp、CNotf Req、および MBReq メッセージで MME から受信した場合の PGW への PRA 情報の受け渡し。
- PRA 情報が MBReq 内に存在する場合は、常にメッセージを PGW に渡します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C は、PGW から受信した指示を MME にリレーし、こうして MME によって報告された情報を PGW にリレーします。

情報を渡すため、*3GPP TS 29274 V15.4.0* に従って GTPC インターフェイスで 2 つの IE が定義されます。次のセクションでは、cnSGW-C に実装されている IE サポートについて説明します。

Presence Reporting Area のアクション

表 *192 : Presence Reporting Area* のアクション

オクテット	ビット							
	8	**7**	**6**	**5**	**4**	**3**	**2**	**1**
1	タイプ = 177							

オクテット	ビット							
	8	7	6	5	4	3	2	1
2 〜 3	長さ = n							
4	SPARE				インスタンス			
5	SPARE				INAPRA	Action		
6 〜 8	Presence Reporting Area 識別子							

表 *193 :* アクション値

Action	値（10 進数）
Presence Reporting Area（PRA）での UE プレゼンスの変更レポートを開始	1
Presence Reporting Area（PRA）での UE プレゼンスの変更レポートを停止	2

- アクション値 1（変更レポートを開始）は、Presence Reporting Area 識別子と、Presence Reporting Area を構成する Presence Reporting Area 要素（存在する場合）によって識別される、Presence Reporting Area での UE プレゼンスの変更レポートを開始するよう要求するために使用されます。
- アクション値 2（変更レポートを停止）は、Presence Reporting Area での UE プレゼンスの変更レポートを停止するよう要求するために使用されます。
- PRA が、MME/SGSN 間のモビリティ手順の間に送信元 MME/SGSN で非アクティブであった場合（つまり、この Presence Reporting Area での UE プレゼンスの変更レポートが PGW/PCRF/OCS によって要求されたが、送信元 MME/SGSN によって非アクティブ化された場合）、オクテット 5 の非アクティブ PRA（IINAPRA）フラグは、S10/S16/S3 インターフェイスで 1 に設定されます。
- アクション値によって Presence Reporting Area での UE プレゼンスの変更レポートの開始、停止、または変更を要求する場合、Presence Reporting Area 識別子が存在する必要があります。その場合、Presence Reporting Area 識別子には Presence Reporting Area の識別子が含まれ、完全な 16 進表記でエンコードされる必要があります。

Presence Reporting Area の情報

表 *194 : Presence Reporting Area* の情報

オクテット	ビット							
	8	**7**	**6**	**5**	**4**	**3**	**2**	**1**
1	タイプ = 178							
2 ～ 3	長さ = n							
4	SPARE				インスタンス			
5 ～ 7	PRA 識別子							
8	SPARE				INAPRA	SPARE	OPRA	IPRA

- オクテット 5 ～ 7 に PRA 識別子が存在し、UE が出入りする PRA の識別子が含まれています。完全な 16 進表記（ASCII エンコードではなくバイナリ）を使用してエンコードされます。PRA 識別子は *3GPP TS 23.003 [2], Clause 19.10* で定義されています。
- UE が PRA 識別子によって識別される Presence Reporting Area の内部にあるか、またはその領域に入ると、内部 PRA（IPRA）フラグが 1 に設定されます。
- UE が PRA 識別子によって識別される Presence Reporting Area の外側にあるか、またはその領域から出ると、外部 PRA（OPRA）フラグが 1 に設定されます。
- PRA が MME/SGSN で非アクティブの場合（つまり、過負荷状態などが原因で、この PRA での UE プレゼンスの変更レポートが MME/SGSN で現在非アクティブになっている場合）、オクテット 8 の非アクティブ PRA（IINAPRA）フラグが 1 に設定されます。
- 特定の Presence Reporting Area 識別子について、IPRA または OPRA フラグまたは INAPRA のいずれか（複数ではなく）が 1 に設定されます。

第 42 章

冗長性サポート

- 機能の概要と変更履歴 （579 ページ）
- 高可用性のサポート （580 ページ）
- ラック間冗長性サポート （583 ページ）

機能の概要と変更履歴

要約データ

表 *195* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効：設定が必要です
関連資料	該当なし

マニュアルの変更履歴

表 *196* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

高可用性のサポート

機能説明

cnSGW-C は Kubernetes クラスタ戦略に基づいて構築されているため、K8 クラスタ展開の高可用性の側面を継承します。cnSGW-C は、ポッドやサービスなどのコンポーネントを含む構造を使用します。

各ポッドには、次に対して高可用性を確保するために少なくとも 2 つのインスタンスがあります。

- ポッドインスタンスの再起動または障害
- ノードの再起動または障害によりポッドが失われました

ポッドとサービスの詳細については、このガイドの「ポッドとサービスの参考資料 （85 ページ）」の章を参照してください。

UDP プロキシの高可用性

cnSGW-C は、UDP プロキシの高可用性（HA）をサポートしています。UDP プロキシの HA モデルは、キープアライブ仮想 IP の概念に基づいています。

UDP プロキシ冗長性に関する情報については、「ポッドとサービスの参考資料（85 ページ）」章の「UDP プロキシの高可用性 （91 ページ）」セクションを参照してください。

アーキテクチャ

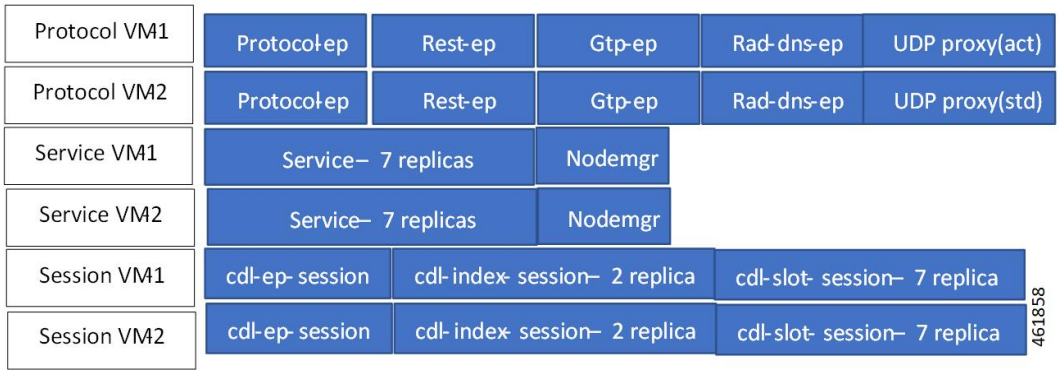
このセクションでは、cnSGW-C ポッドと VM の推奨レイアウトについて説明します。

cnSGW-C ポッドおよび VM の展開レイアウト

ここでは、cnSGW-C ポッドとそのマイクロサービスの展開について説明します。

次の図は、cnSGW-C における 6 つの VM の導入モデルを示しています。

図 *108 : VM* 導入モデル



このモデルでは、ポッドはVMペアに展開されます。各プロトコルポッド（rest-ep、protocol-ep、gtp-ep など）に対して 2 つのレプリカを使用できます。各プロトコル VM に 1 つのインスタンスが展開されます。

同様に、サービスポッドとセッションポッドは、サービスVMとセッションVMの両方に均等に分散されます。このような分散は、VMにラベルを付け、ポッドのスケジューリング中にK8アフィニティルールとアンチアフィニティルールを実装することによって制御されます。

このモデルにより、VM の再起動シナリオ中に、各 ポッドタイプの少なくとも 50% のレプリカがユーザーシグナリングの処理に使用可能になります。

ポッドのグレースフル再起動により、ポッドは進行中の処理を30秒以内に完了できます。ポッドの突然の再起動は、PDUセッションに影響を与えずに、進行中のトランザクションに影響を与えます。

機能の仕組み

ここでは、復元力と HA を実現する方法について説明します。

cnSGW-C は、ポッドの障害中または再起動中にポッド間通信を可能にします。

ポッドの正常な再起動中：

- 進行中の処理は影響を受けません。
- 新しいメッセージは、Kubernetes サービスを介してポッドに送信されません。
- セッションアフィニティを持つメッセージは、引き続きポッドで受信されます。
- 既存のコールフローは 30 秒以内に完了することが予想されます。

ポッドの再起動後：

- ポッドのすべての Prometheus メトリックがリセットされます。
- 内部診断が緑色の場合、ポッドのステータスは [Ready] に変わります。
- ポッドは新しいメッセージを処理する準備ができています。

cnSGW-C VM が再起動する、または VM が使用できない場合：

- VM 上のすべてのポッドが失われます。
- 使用可能な他の VM 上のポッドは処理を継続するため、高可用性が実現します。
- VIP（存在する場合）は、他の使用可能なノードに切り替えられます。
- Kubernetes がノードをダウン状態として検出するには、ノードが到達不能になってから約 5 分間かかります。
- その後、ノード上のポッドは、Kubernetes サービスでは検出できなくなります。

ポッドが再起動すると、VM 上のポッドが順次スケジュールされます。この操作は、ポッドの再起動に似ています。

VIP と VM の再起動中に、仮想 IP が単一の VM に関連付けられます。UDP プロキシは、UPF との通信用に N4 VIP アドレスにバインドします。UDP プロキシは、cnSGW-C との通信用に S5 VIP アドレスにバインドします。

アクティブな VIP で VM を再起動すると、VIP は他のプロトコルの VM に切り替わります。アクティブな UDP プロキシに障害が発生すると、VIP は他のプロトコルの VM に切り替わります。

サブスクライバ マイクロサービス インフラストラクチャ（SMI）が VIP のモニタリングとスイッチオーバーを処理する前に、SMI 展開で適切な VIP 構成が使用可能であることを確認してください。また、ポートが 28000 に設定されており、ホストの優先順位が同じであるかどうかを確認します。

ポッドレベルのラベル付けとレプリカの構成

ノードラベルは、SMI クラスタデプロイヤで構成されます。構成コマンドに関する詳細は、「オペレーションセンターでの cnSGW-C の展開と構成 （19 ページ）」章の「ノードラベルを持つマッピングポッド （21 ページ）」セクションを参照してください。

設定例

次に、VM のラベル付けとレプリカの設定例を示します。

```
k8 label protocol-layer key smi.cisco.com/node-type value smf-proto
exit
k8 label service-layer key vm-type value smf-svc
exit
k8 label cdl-layer key smi.cisco.com/node-type value smf-cdl
exit
k8 label oam-layer key smi.cisco.com/node-type value oam
exit

endpoint pfcp
 replicas 1
 nodes    2
exit
endpoint service
 replicas 1
 nodes    2
```

```
exit
endpoint protocol
 replicas 1
 nodes    2
 vip-ip 209.165.201.28
exit
endpoint sbi
 replicas 1
 nodes    2
```

設定の確認

構成を確認するには、次の show コマンドを使用します。

show running-config instance instance-id *instance_id* **endpoint**

以下は、この show コマンドの出力例です。

```
show running-config instance instance-id 1 endpoint
instance instance-id 1
 endpoint nodemgr
  replicas 1
  nodes    2
 exit
 endpoint gtp
  replicas 1
  vip-ip 209.165.201.29
 exit
 endpoint pfcp
  replicas                2
  enable-cpu-optimization true
  interface n4
   heartbeat
    interval               0
    retransmission-timeout 3
    max-retransmissions    5
   exit
  exit
 exit
 endpoint service
  replicas 2
 exit
 endpoint protocol
  replicas 1
  vip-ip 209.165.201.29
  end
```

このコマンド出力には、エンドポイント名、ポッドレプリカ、ノードなど、複数のエンドポイントに関連する構成が表示されます。

ラック間冗長性サポート

ラック間冗長性サポートとは、1 つのラックで障害や停止が発生した場合でも、システムまたはサービスがその機能と可用性を維持する機能を意味します。これは、同じ地理的な場所にある別のラックに操作を移動することで軽減できます。

機能説明

cnSGW-C、アクティブ-アクティブモードのラック間冗長性をサポートします。ラック間の冗長性は、リモートラックへのサービスのシームレスなフェールオーバーとフェールバックに必要なセッション、構成、およびその他のデータのレプリケーションによって実現されます。

機能の仕組み

cnSGW-C（CNF）を同じデータセンターに展開して、SMF cnSGW-C クラスタをホストするラックに対してローカライズされた、致命的な障害向けのサービスを提供できます。

各 CNF インスタンスのサービスは、MME/SGW の DNS エントリのために NRF および S11/S5 に登録されます。ローカル HA 冗長性により、インスタンスは、同じデータセンター内の K8 クラスタレベルの障害に加えて、ラックレベルの冗長性を実現したり、障害のあるコンテナがタイプ 2 < *n* に従う場合に、同じ K8 クラスタ内でローカルに処理したりできます。

n は値です。コンテナ障害が 50% 未満の場合は、HA で障害を処理する必要があります。コンテナ障害が 50% を超えると、ラック間スイッチオーバーがトリガーされます。

概要

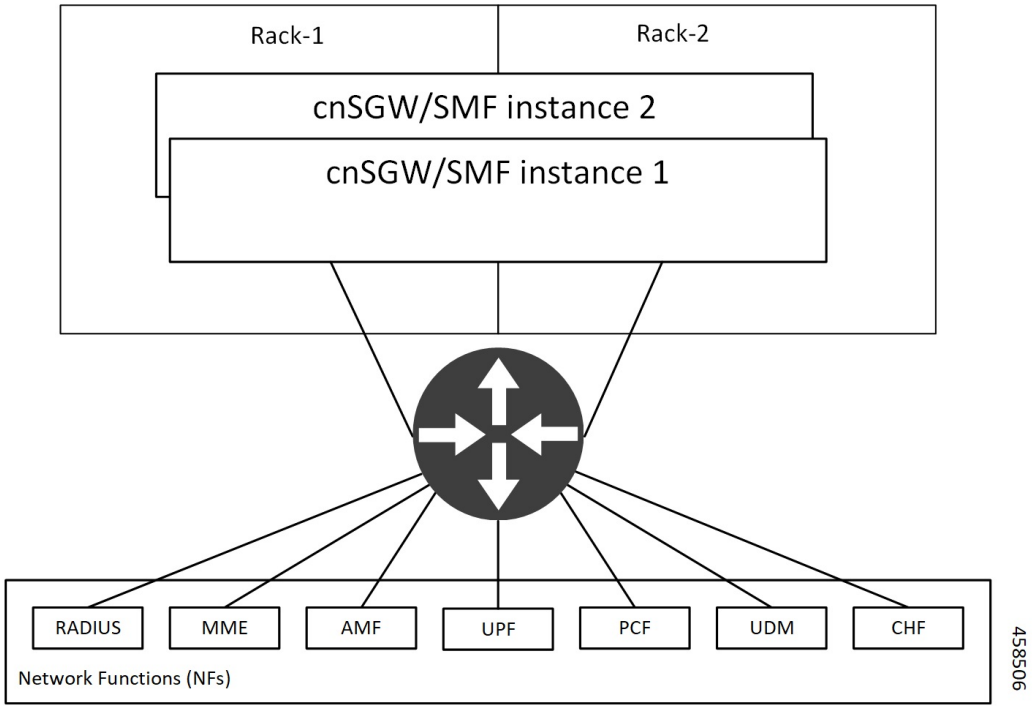
アクティブ-アクティブモード

- ラック間の展開は、隣接する NF に対して透過的です。
- ラック間の展開には CCG 機能の 2 つのインスタンスが含まれ、各インスタンスは一連のインターフェイス IP で識別されます。
- 各インスタンスは一連のセッションをサポートし、セッションの一貫性のために同じ IP を使用し続けます。
- 特定の期間において、1 つの CCG インスタンスは一方のラックでのみプライマリとして動作し、もう一方のラックではスタンバイとして動作します。
- CCG インスタンスに関連付けられている一連のインターフェイス IP は、インスタンスのプライマリラックに動的にルーティングされます。

cnSGW-C は、データがプライマリからスタンバイインスタンスに複製されるプライマリ/スタンバイ冗長性をサポートします。通常の動作では、プライマリインスタンスがサービスを提供します。プライマリインスタンスで障害が発生すると、スタンバイインスタンスがプライマリになり、動作を引き継ぎます。ラック間の冗長性を実現するために、2つのプライマリ/スタンバイペアを設定できます。この場合は各ラックがアクティブにトラフィックを処理し、スタンバイ側がリモートラックのバックアップとして機能します。

アクティブ-アクティブのラック間冗長性展開で、同じデータセンターに 2 つのラックがあるとします（ラック 1 と ラック 2）。すべての NF がインスタンス 1 とインスタンス 2 に到達しようとしています。

図 *109* : アクティブ-アクティブのラック間冗長性展開



NF の場合、両方のインスタンスがアクティブになります。しかし実際には、インスタンス 1 とインスタンス 2 はラックにまたがって配置されます。

ラック 1 にはインスタンス 1 とインスタンス 2 があります。トリガー前のシナリオでは、インスタンス 1 はローカルでプライマリとして機能し、インスタンス 2 はスタンバイモードになります。

ラック 2 にもインスタンス 1 とインスタンス 2 があります。トリガー前のシナリオでは、インスタンス 2 はローカルでプライマリとして機能し、インスタンス 1 はスタンバイモードになります。

ラック 1 に障害が発生すると、トラフィックはラック 2 に移動します。ラック 2 では、インスタンス 1 とインスタンス 2 の両方がプライマリとして機能します。

ラック間冗長性トリガー

ラック間冗長性は、次のトリガーをサポートしています。

- **CLI ベースのスイッチオーバー**：手動の CLI コマンドを使用して、ロールを切り替え、ラック間冗長性フェールオーバーをトリガーします。
- **BFD リンクフェールオーバー検出**：接続されたラックとリーフ間の両方の BFD リンクがダウンすると、ラック間冗長性フェールオーバーがトリガーされます。

- **ローカルラックポッド障害検出**：ポッドレプリカセットの障害のしきい値パーセンテージが、設定されたしきい値を上回ると、ラック間冗長性フェールオーバーがトリガーされます。
- **リモートラックポッド障害検出**：リモートポッドモニタリングでしきい値パーセンテージを超える障害が検出されると、ポッドはそのインスタンスに対してセルフプライマリになります。
- **リモート ラック ロール モニタリング**：リモートロールモニタリングで、ラックがStandy_error状態であることが検出されると、そのラックはセルフプライマリになります。
- **マルチコンピューティング障害**：2 つ以上のサーバーの電源がオフになると、ラック間冗長性フェールオーバーがトリガーされます。

ラック NF ロール

次に、適用可能なラック NF ロールのリストを示します。

（注）
- **Cachepod/etcd** および **CDL レプリケーション**は、次のセクションで説明されているすべてのロール中に発生します。
- ラック間リンクがダウンしているか、定期的なハートビートに失敗すると、これらのラック間冗長性トリガーが一時停止します。

- **PRIMARY**：このロールでは、ラックは準備完了状態で、特定のインスタンスのトラフィックをアクティブに取得しています。
- **[STANDBY]**：このロールでは、ラックはスタンバイモードであり、トラフィックを取得する準備ができていますが、特定のインスタンスのトラフィックを取得しません。
- **STANDBY_ERROR**：このロールでは、ラックは問題のある状態で、アクティブではなく、特定のインスタンスのトラフィックを取得する準備ができていません。

（注）
インスタンスロールが **STANDBY_ERROR** の状態の場合、データの複製は停止します。この状態では、**show georeplication-status** コマンドは常に失敗します。ただし、インスタンスのロールが **STANDBY** に移行されると、データの複製が自動的に再開され、コマンドは結果が **pass** であると表示します。

- **FAILOVER_INIT**：このロールでは、ラックはフェールオーバーを開始し、トラフィックを取得する状態ではありません。アプリケーションがアクティビティを完了するまでのバッファ時間は 2 秒です。

- **FAILOVER_COMPLETE**：このロールでは、ラックはフェールオーバーを完了し、特定のインスタンスのフェールオーバーについてピアラックに通知しようとしています。バッファ時間は 2 秒です。
- **FAILBACK_STARTED**：このロールでは、手動フェールオーバーは、特定のインスタンスのリモートラックから遅延してトリガーされます。

新規インストールの場合、ラックは次のロールでブートアップします。

- **PRIMARY**：このロールでは、ラックはローカルインスタンス用です（各ラックには、ローカルインスタンスを識別するように構成されたローカル**instance-id**があります）。新規インストール中は、ポッドをモニタリング用に構成しないことをお勧めします。セットアップの準備ができたら、モニタリング用にポッドを構成できます。
- **STANDBY**：このロールでは、ラックは他のインスタンスの にあります。

アップグレードの場合、ラックは次のロールでブートアップします。

- **STANDBY_ERROR**：このロールでは、アップグレード後にトラフィックを移動させるには手動の介入が必要なため、すべてのインスタンスがラックになります。
- **ETCD**：このロールでは、ラックはインスタンスのロールを保存します。

（注） ローリング アップグレードまたはインサービスアップグレードはサポートされていません。

一般的な注意事項

ラック間冗長性展開を設定する前に、次の一般的なガイドラインを参照してください。

- 両方のラックのソフトウェアバージョンが同じである必要があります。
- 両方のラックの設定を同じにする必要があります。
- インスタンス 1 とインスタンス 2 のループバック ポートは異なる必要があります。同じ場合、K8 IP/ ポートの競合が原因で REST-EP ポッドが起動しません。
- 両方のラックの各インターフェイスは、同じ VLAN 上にある必要があります。たとえば、インスタンス 1 とインスタンス 2 の N4 VLAN は、同じ VLAN 上にある必要があります。異なる場合、BGP ポリシーの適用時にカーネルでルートの競合が発生します。
- 適切なロールを割り当てるために、シスコのテクニカル担当者に問い合わせて、次の手順を実行します。

  詳細については、GR ペアのソフトウェアアップデート （616 ページ）を参照してください。
- フェールオーバー後、ラックが正常であることを確認してから、フェールバックを手動で実行します。自律フェールバックはサポートされていません。

詳細については、回復手順 （639 ページ）を参照してください。

- BGP ピアリングには、BGP スピーカーポッドで非結合インターフェイスを使用します。
- Proto ノードごとの BGP ピアリングは、2 つの BGP ルータ/リーフのみでサポートされます。2 つの Proto ノードを考えている場合、最大 4 つの BGP ネイバーシップを設定できます。
- サービストラフィックには結合インターフェイスを使用します。
- Geo ポッドでは、次の 2 つの VIP を使用します。
  - ポッド間通信用の内部 VIP（ラック内）
  - ラック間 Geo ポッド通信用の外部 VIP。L2 サブネットの Proto ノードでのみ設定します。これはラック間の通信に使用されます。このノードには他のラックへの外部接続があります。
- ラック内のすべてのノードに到達可能な Geo 内部 IP。
- Geo 外部 IP：
- CDL/Kafka VIP：L2 サブネット上の CDL ラベル付きノードで設定します。
- 両方のラックで LI タップを有効にします。
- MDF サーバーは、両方のラックから到達可能である必要があります。

インスタンス認識

cnSGW-C でのインスタンス認識構成は、ローカルラックインスタンスとリモートラックインスタンスを区別するのに役立ちます。

ラック間冗長性インスタンスの構成

この設定は、複数ラックのラック間冗長性構成を提供するために必要です。インスタンス ID を使用して、ラックごとにエンドポイントを構成する必要があります。

構成例 1

以下は、1 つのインスタンスにおけるエンドポイント VIP 構成の構成例です。

```
config
  instance instance-id gr_instanceId
    endpoint endpoint_name
        vip-ip vip_ip_address
  exit
exit
```

例：

```
config
instance instance-id 1
```

```
 endpoint sbi
  vip-ip 209.165.201.21
 exit
exit
```

構成例 2

次に、インスタンスのシステム ID、クラスタ ID、スライス名に関する情報を提供する構成例を示します。

```
config
  instances instance instance_id
   system-id system_id
   cluster-id cluster_id
   slice-name cdl_slice_name
  exit
exit
```

例：

```
config
 instances instance 1
  system-id sgw
  cluster-id sgw
  slice-name 1
 exit
exit
```

（注） インスタンスの *system-id*、*cluster-id*、および展開の *app-name*、*cluster-name* に同じ値を使用することをお勧めします。

エンドポイントインスタンス認識の構成

ローカルラックとリモートラックごとに2つのインスタンスのみを構成でき、対応するエンドポイントをインスタンス化できます。

ローカルインスタンス識別子は、ラックが冗長かどうかに関係なく、ローカルラックの識別子です。

ローカルインスタンス識別子の設定

ローカルインスタンスは、 local-instance コマンドを使用して構成されます。

```
local-instance instance 1
```

エンドポイント構成は、各一意のインスタンス識別子で指定されたインスタンスの下にある必要があります。

エンドポイントの構成例

次の構成例を参照してください。

（注） 次の例では、インスタンス識別子「**1**」がローカルインスタンス識別子であり、その下に構成されたエンドポイントはローカルサイトに属しています。

必要に応じて、リモートラックインスタンス識別子「**2**」をラックに属するエンドポイントに構成できます。

```
instance instance-id 1
 endpoint li
  replicas 1
  nodes    2
  vip-ip 209.165.201.6
  vip-ip 209.165.201.13
 exit
 endpoint gtp
  replicas 1
  nodes    2
  retransmission timeout 5 max-retry 4
  vip-ip 209.165.201.6
  vip-ip 209.165.201.4
  interface s5
   echo interval          60
   echo retransmission-timeout 5
   echo max-retransmissions 4
  exit
  interface s2b
   echo interval 60
   echo retransmission-timeout 5
   echo max-retransmissions 4
  exit
 exit
exit
instance instance-id 2
 endpoint li
  replicas 1
  nodes    2
  vip-ip 209.165.201.6
  vip-ip 209.165.201.13
 exit
exit
endpoint gtp
  replicas 1
  nodes    2
  retransmission timeout 5 max-retry 4
  vip-ip 209.165.201.6
  vip-ip 209.165.201.5
  interface s5
   echo interval 60
   echo retransmission-timeout 5
   echo max-retransmissions 4
  exit
  interface s2b
   echo interval 60
   echo retransmission-timeout 5
   echo max-retransmissions 4
  exit
 exit
exit
```

プロファイル cnSGW-C インスタンス認識の構成

ローカルおよびリモートインスタンスに対応する PGW FQDN のインスタンスを追加します。

例

次に、構成例を示します。

（注） 次の例では、*instance-id*「**1**」がローカルインスタンス ID であり、その下に構成された cnSGW-C プロファイルがローカルラックに属しています。

必要に応じて、リモートラック インスタンス *ID*「**2**」をラック間に属する FQDN に構成できます。

```
profile sgw sgw1
locality LOC1
instances 1 fqdn cisco.com.apn.epc.mnc456.mcc123
instances 2 fqdn cisco.com.apn.epc.mnc567.mcc123
```

cnSGW-C エンドポイントの設定

エンドポイントの設定は、cnSGW-C にのみ必要です。

例

次に、設定例を示します。

（注） 次の例では、instance-id「**1**」がローカルインスタンス ID で、その下に設定されたエンドポイントはローカルサイトに属しています。

必要に応じて、リモートサイト instance-id「**2**」を地理的サイトに属するエンドポイントに設定できます。

```
instance instance-id 1
 endpoint nodemgr
  replicas 1
  nodes    1
 exit
 endpoint gtp
  replicas 1
  vip-ip 209.165.201.10
  interface s5e
   vip-ip 209.165.201.29
  exit
  interface s11
   vip-ip 209.165.201.29
  exit
 exit
 endpoint pfcp
  replicas 1
  interface sxa
   heartbeat
```

```
    interval               0
    retransmission-timeout 5
    max-retransmissions    3
   exit
  exit
exit
 endpoint service
  replicas 1
 exit
 endpoint protocol
  replicas 1
  vip-ip 209.165.201.29
  interface sxa
   vip-ip 209.165.201.29
  exit
 exit
 endpoint sgw-service
  replicas 1
 exit
exit
instance instance-id 2
 endpoint nodemgr
  replicas 1
  nodes    1
 exit
 endpoint gtp
  replicas 1
  vip-ip 209.165.202.150
  interface s5e
   vip-ip 209.165.201.27
  exit
  interface s11
   vip-ip 209.165.201.27
  exit
 exit
 endpoint pfcp
  replicas 1
  interface sxa
   heartbeat
    interval               0
    retransmission-timeout 5
    max-retransmissions    3
   exit
  exit
exit
 endpoint service
  replicas 1
 exit
 endpoint protocol
  replicas 1
  vip-ip 209.165.201.27
  interface sxa
   vip-ip 209.165.201.27
  exit
 exit
 endpoint sgw-service
  replicas 1
 exit
exit
```

ダイナミック ルーティング

ボーダー ゲートウェイ プロトコル（BGP）では、自律システム（AS）間にループフリーのドメイン間ルーティングを作成可能です。AS は、単一の技術管理に基づくルータのまとまりです。ルータは、外部ゲートウェイプロトコルを使用して AS の外部にパケットをルーティングできます。BGP を使用したダイナミックルーティング機能を使用すると、優先順位とルートとともに、IP アドレスにサービスを提供する代替ローカルアドレスを持つ BGP ルータのネクストホップ属性を構成できます。App-Infra BGP スピーカーポッドは、BGP を使用してポッドルートをサービス VIP にアドバタイズすることにより、トラフィックのダイナミックルーティングを可能にします。

この機能は、次の機能をサポートしています。

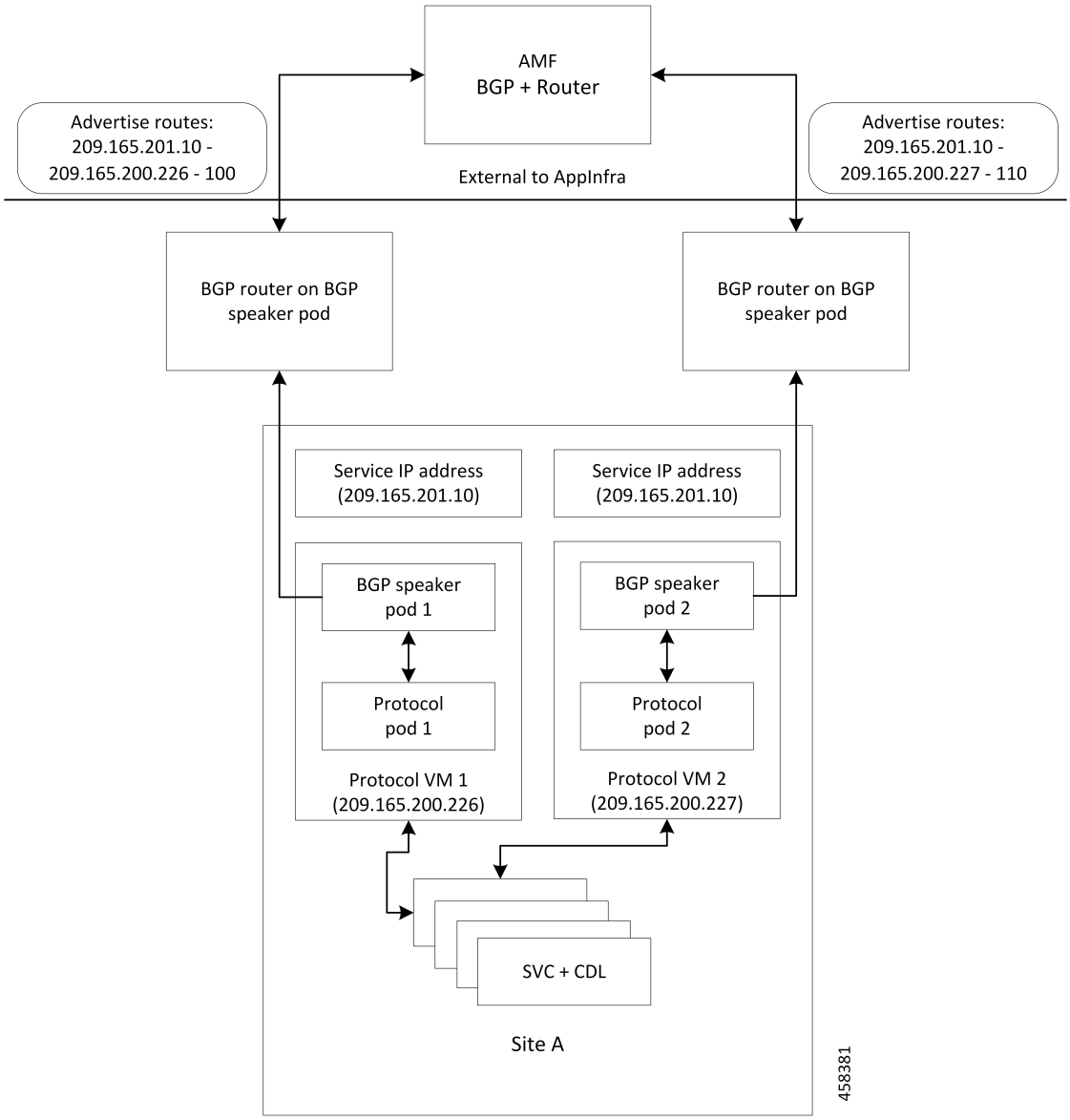
- BGP を使用したダイナミックルーティングによる着信トラフィックのサービス IP アドレスのアドバタイズ。
- 発信トラフィックのルートの学習。
- BGP ポッドフェールオーバーの処理。
- プロトコル ポッド フェールオーバーの処理。
- BGP スピーカーの統計と KPI。
- BGP スピーカーをデバッグするためのログメッセージ。
- BGP スピーカーポッドの有効化または無効化。
- BGP を設定するための新しい CLI コマンド。

着信トラフィック

BGP はトランスポート プロトコルとして TCP（ポート 179）を使用します。2 台の BGP ルータは相互に TCP 接続を形成します。それらのルータはピアルータです。ピアルータはメッセージを交換し、接続パラメータを開いて確認します。

BGP スピーカーは、アクティブ/スタンバイモードで着信トラフィックのプロトコルポッドのルーティング情報を公開します。次の図を例として使用して、ダイナミックルーティングの機能を理解してください。pod1 と pod2 の 2 つのプロトコルポッドがあります。Pod1 はアクティブで、pod2 はスタンバイモードです。サービス IP アドレス 209.165.200.225 は、209.165.200.226 と 209.165.200.227 の両方のノードで構成されています。pod1 はホスト 209.165.200.226 で、pod2 はホスト 209.165.200.227 で実行されています。ホスト IP アドレスはポッドサービスを公開します。BGP スピーカーは、ルート 209.165.200.225、209.165.200.226 および 209.165.200.227 に公開します。また、ポッドの優先順位を決定するために、プリファレンス値 110 と 100 を公開します。

図 *110 :* アクティブ/スタンバイ トポロジでの着信トラフィックのダイナミックルーティング



高可用性を実現するため、各クラスタにはアクティブ-スタンバイトポロジを備えた2つのBGPスピーカーポッドがあります。カーネルルートの変更は、プロトコルポッドが実行されているホストネットワークレベルで実行されます。

MED 値

ローカルプリファレンスはIGPネイバーでのみ使用され、MED属性はEGPネイバーでのみ使用されます。BGPでは、より低いMED値が推奨されます。

表 *197 : MED* 値

ボンディングインターフェイスがアクティブ	VIP の存在	MED 値	Local Preference
対応	対応	1210	2220
対応	非対応	1220	2210
非対応	対応	1215	2215
非対応	非対応	1225	2205

BGP スピーカーポッドのブートストラップ

次の一連の手順で、BGP スピーカーポッドをセットアップします。

1. BGP スピーカーポッドは、ポート 179 で、トランスポートプロトコルとして TCP を使用します。これらのポッドは、Ops Center CLI で構成された AS 番号を使用します。
2. Topology Manager を登録します。
3. リーダーポッドを選択します。アクティブなスピーカーポッドがデフォルトの選択肢です。
4. Ops Center CLI によって提供されるすべての BGP ピアへの接続を確立します。
5. ETCD から既存のすべてのルートを公開します。
6. CLI 設定を使用して、ルーティングのインポートポリシーを設定します。
7. 両方のスピーカーポッドで gRPC ストリームサーバーを開始します。
8. キャッシュポッドと同様に、各名前空間で 2 つの BGP スピーカーポッドを実行する必要があります。

動的ルーティングの詳細については、『*UCC* サービング ゲートウェイ コントロール プレーンの機能：構成および管理ガイド』の「*BGP* を使用したダイナミックルーティング」の章を参照してください。

BGP を使用したダイナミックルーティングの設定

ここでは、BGP を使用したダイナミックルーティングの設定方法について説明します。

AS と BGP ルータの IP アドレスの設定

BGP ルータの AS と IP アドレスを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。

  ラック間冗長性展開では、2 つの自律システム（AS）を設定する必要があります。

  - リーフとスパイン用に 1 つの AS。
  - 両方のラックの第 2 AS：Rack-1 と Rack-2。

BGP サービスリスニング IP アドレスの設定

BGP サービスリスニング IP アドレスを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      interface interface_name
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **interface** *interface_name*：インターフェイスの名前を指定します。

BGP ネイバーの設定

BGP ネイバーを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      interface interface_name
      neighbor neighbor_ip_address remote-as as_number
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **interface** *interface_name*：インターフェイスの名前を指定します。
- **neighbor** *neighbor_ip_address*：ネイバー BGP ルータの IP アドレスを指定します。
- **remote-as** *as_number*：AS の ID 番号を指定します。

ボンディング インターフェイスの設定

インターフェイスに関連するボンディングインターフェイスを設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
```

```
      interface interface_name
      bondingInterface interface_name
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **interface** *interface_name*：インターフェイスの名前を指定します。
- **bondingInterface** *interface_name*：インターフェイスに関連するボンディング インターフェイスを指定します。ボンディング インターフェイスがアクティブである場合、BGP は、より低い MED 値を提供することで、インターフェイスサービスに高いプリファレンスを与えます。

デフォルトルートの学習の設定

システム上に特定のルートを設定し、すべてのルートをサポートする必要があるユーザーは、**learnDefaultRoute** を **true** に設定する必要があります。

（注） この設定は任意です。

デフォルトルートの学習を設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      learnDefaultRoute true/false
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **learnDefaultRoute** *true/false*：**learnDefaultRoute** パラメータを有効または無効にするオプションを指定します。true に設定すると、BGP はデフォルトルートを学習して、カーネルスペースに追加します。デフォルトは False です。

BGP ポートの設定

BGP サービスのポート番号を設定するには、次のコマンドを使用します。

```
config
   router bgp local_as_number
      loopbackPort port_number
   exit
exit
```

注：

- **router bgp** *local_as_number*：BGP ルータの AS の ID 番号を指定します。
- **loopbackPort** *port_number*：BGP サービスのポート番号を指定します。デフォルト値は 179 です。

ポリシーの追加

BGP スピーカーポッドは、ネイバーから多くのルート情報を学習しますが、発信トラフィックのサポートに使用されるルートはごく一部のルートです。これは、cnSGW-C が外部の AMF/PCF に情報を送信している場合、出力トラフィックの処理にのみ必要です。ルートは、BGP スピーカーでインポートポリシーを設定することでフィルタリングされ、学習したルートをプロトコルポッドに送信するために使用されます。

ポリシー追加用のサンプル CLI コードと、各パラメータに対応する説明を以下に示します。

```
$bgp policy <policy_Name> ip-prefix 209.165.200.225 subnet 16 masklength-range 21..24
as-path-set "^65100"
```

表 *198*：インポートポリシーのパラメータ

要素	説明	例	オプション
as-path-set	AS パス値	“^65100”	対応
ip-prefix	プレフィックス値	“209.165.200.225/16”	対応
masklength-range	長さの範囲	“21..24”	対応
interface	送信元 IP として設定するインターフェイス（デフォルトは VM IP）	eth0	対応
gateWay	着信ルートのゲートウェイを変更します。	209.165.201.30	対応
modifySourceIp	着信ルートの送信元IPを変更します。 デフォルト値は False です。	true	対応
isStaticRoute	カーネルルートに静的IPアドレスを追加するフラグ デフォルト値は False です。	true	対応

BGP スピーカーの設定

この設定は、展開内の BGP スピーカーポッドの数を制御します。BGP スピーカーは、両方のラックからの着信トラフィックのサービス IP 情報をアドバタイズします。

（注）
- BGP ピアリングには、BGP スピーカーポッドで非結合インターフェイスを使用します。
- Proto ノードごとの BGP ピアリングは、2 つの BGP ルータ/リーフのみでサポートされます。2 つの Proto ノードを考慮すると、最大 4 つの BGP ネイバーシップを設定できます。

```
instance instance-id  instance_id endpoint bgpspeaker interface { bgp |
bfd } internal base-port start base_port_number
```

```
config
instance instance-id instance_id
 endpoint bgpspeaker
   replicas replica_id
   nodes node_id
   interface bgp
     internal base-port start base_port_number
   exit
   interface bfd
     internal base-port start base_port_number
  exit
exit
```

注：

- **instance instance-id** *instance_id*：GR インスタンス ID を指定します。
- *base_port_number*：論理 NF が設定されている場合にのみ、ポート範囲を指定します。この範囲は、展開によって異なります。

例

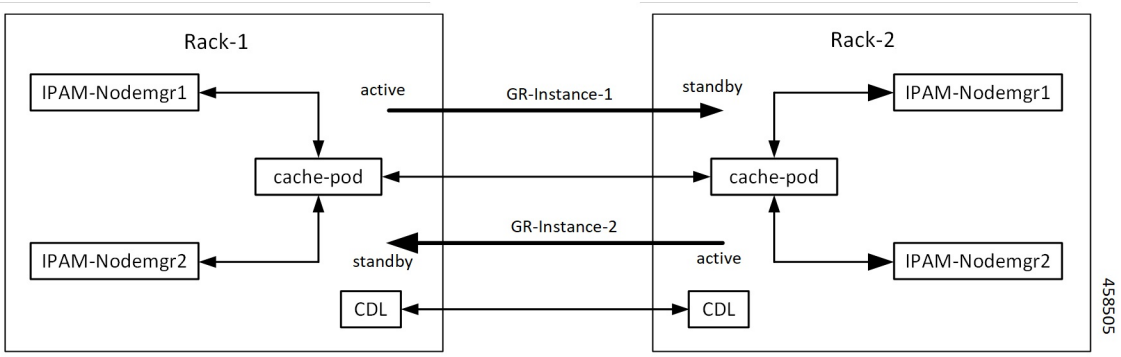
次に、設定例を示します。

```
instance instance-id 1
endpoint bgpspeaker
   replicas 1
   nodes    2
   interface bgp
     internal base-port start {24000}
   exit
   interface bfd
     internal base-port start {25000}
 exit
```

IPAM

ここでは、ラックレベルでの IP アドレス管理（IPAM）について説明します。

図 *111 : IPAM*



UPF 登録時、アクティブな IPAM インスタンスが各 DNN の UPF ごとに 4 つのアドレス範囲を予約します。

- 範囲 1：アクティブクラスタ、nodemgr-1
- 範囲 2：アクティブクラスタ、nodemgr-2
- 範囲 3：スタンバイクラスタ、nodemgr-1
- 範囲 4：スタンバイクラスタ、nodemgr-2

通常の動作中、Rack-1 は GR-instance-1 で起動するサブスクライバの UPF 登録/リリース、アドレス割り当て/リリースを処理します。

Rack-2 がダウンすると、Rack-1 が GR-Instance-2 のロール変更トリガーを取得します。

- Rack-1 の IPAM は、local-cache-pod（すでに同期済み）から GR-Instance-2 のコンテンツを復元します。
- Rack-1 の IPAM は、GR-Instance-1 の処理に加えて、復元されたコンテンツを使用して GR-Instance-2 で起動するサブスクライバの UPF 登録/リリースおよびアドレス割り当て/リリースを処理します。

各 IPAM プールは、次のように GR インスタンスに関連付けられます。

- プール名は、すべてのインスタンス間で一意です。
- アドレス範囲は、VRF 内およびすべてのインスタンス間で一意です。

特定のインスタンスのアクティブとスタンバイの両方の cnSGW-C クラスタで、同じプール設定を設定する必要があります。

アドレス割り当て中に、アクティブなインスタンスは UPF 用に予約されたアドレス範囲から空き IP を割り当てます。

新しいアドレス範囲を使用できない場合は、スタンバイのアドレス範囲の所有権を現在のアクティブインスタンスに変更して、アドレス範囲の割り当てを続行します。

IPAM の構成

次のセクションでは、IPAM の構成例を示します。

cnSGW-C-1 の例

cnSGW-C-1 の構成例を次に示します。

```
ipam
 instance 1
  address-pool pool-1
  vrf-name ISP
   tags
   dnn dnn-1
  exit
  ipv4
    address-range 209.165.201.1 209.165.201.31
 exit
instance 2
 address-pool pool-2
  vrf-name ISP
  tags
   dnn dnn-2
  exit
  ipv4
    address-range 209.165.202.129 209.165.202.159
 exit
exit
```

cnSGW-C-2 の例

cnSGW-C-2 の構成例を次に示します。

```
ipam
 instance 1
 address-pool pool-1
  vrf-name ISP
  tags
   dnn dnn-1
  exit
  ipv4
    address-range 209.165.201.1 209.165.201.31
 exit
 instance 2
 address-pool pool-2
  vrf-name ISP
   tags
   dnn dnn-2
  exit
  ipv4
    address-range 209.165.202.129 209.165.202.159
 exit
exit
```

Geo レプリケーション

Geo レプリケーションはラック間通信で、ラック内の POD、VIP、または BFD のモニタリングで使用されます。地理的冗長性は、次で構成されます。

- ラックごとに Geo ポッドの 2 つのインスタンスが実行されていること。
- 2 つの Geo ポッドがアクティブ/スタンバイモードで機能する。
- 各 Geo ポッドインスタンスは、異なる Proto ノードまたは VM で生成されます。
- Proto ノードまたは VIP を持つ VM で実行されている Geo ポッドがアクティブな Geo ポッドです。
- アクティブな Geo ポッドが再起動した場合、VIP は他の Proto ノードまたは VM に切り替えられ、他の Proto ノードまたは VM で実行されているスタンバイ Geo ポッドがアクティブになります。
- 地理ポッドは、ホスト ネットワーキング モードを使用します（UDP プロキシと同様）。
- Geo ポッドでは、次の 2 つの VIP を使用します。
  - **内部**：ポッド間通信用の VIP（ラック内）
  - **外部**：ラック間 Geo ポッド通信用の VIP

    L2 サブネット上の Proto ノードでのみ構成されます。ラック間の通信に使用されます。このノードには他のラックへの外部接続があります。

- Logical-NF-InstanceID は、GR-Pair の両方の cnSGW-Cs で同じに設定する必要があります。
- KeepAliveD モニタリングの場合：
  - 地理ポッドは、15000+（Logical-NF-InstanceID * 32）+4 としてベースポートを使用します。

    Geo ポッドのベースポートは、BGP スピーカーのポッドポートとは異なる必要があります。
    - デフォルトポート（論理 cnSGW-Cなし）：15004
    - Logical-nf-instance-id が 1 に構成され、ポートが 15036 に構成された論理 cnSGW-C の場合

  - UDP プロキシポッドは、ベースポートを 28000+Logical-NF-InstanceID として使用します。
    - デフォルトポート（論理 cnSGW-Cなし）：28000
    - Logical-nf-instance-id が 1 に構成され、ポートが 28001 に構成された論理 cnSGW-C の場合

  - BGP スピーカーポッドは、デフォルトのベースポートを 20000+(Logical-NF-InstanceID * 32)+4 として使用します。
    - デフォルトポート（論理 cnSGW-Cなし）：20004
    - 論理的 cnSGW-C の場合、logical-nf-instance-id が 1 に構成され、ポートが 20036 に設定される

（注） ETCD およびキャッシュポッドデータのみがスタンバイラックに複製されます。

ETCD/CachePod レプリケーションの設定

エンドポイントは、インスタンスの下に設定する必要があります。各ラックには2つの地理的冗長性ポッドが必要です。また、ETCD/CachePod レプリケーショのために、内部および外部の地理的インターフェイスに VIP を設定する必要があります。

instance instance-id *instance_id* **endpoint geo interface { geo-internal | geo-external } vip-ip {** *vip_ip_address* **} vip-port {** *vip_port_number* **}**

```
config
instance instance-id instance_id
 endpoint geo
  replicas replica_id
  nodes node_id
  internal base-port start base_port_number
  interface geo-internal
   vip-ip vip_ip_address vip-port vip_port_number
  exit
  interface geo-external
   vip-ip vip_ip_address vip-port vip_port_number
  exit
exit
exit
```

注：

- **instance instance-id** *instance_id*：GR インスタンス ID を指定します。1 つのインスタンス ID はローカルラック用、もう 1 つは別のラック用です。
- **vip-ip** *vip_ip_address*：内部/外部の地理的インターフェイスの VIP IP アドレスを指定します。
- **vip-port** *vip_port_number*：VIP ポート番号を指定します。
- **internal base-port start** *base_port_number*：論理 NF が設定されている場合にのみ、ポート範囲を指定します。

例

次に、設定例を示します。

```
instance instance-id 1
endpoint geo
  replicas 1
  nodes 2
  internal base-port start 25000
  interface geo-internal
   vip-ip 209.165.201.8 vip-port 7001
  exit
  interface geo-external
```

```
   vip-ip 209.165.201.8 vip-port 7002
  exit
exit
```

Geo モニタリング

ここでは、Geo モニタリングについて説明します。

ポッドモニタリング

ラック間セットアップでポッドモニタリングおよびフェールオーバーしきい値を設定するには、次の設定例を使用します。Geo ポッドは、設定されたポッド名をモニターします。

```
config
 geomonitor
  podmonitor pods pod_name
   retryCount value
   retryInterval interval_value
   retryFailOverInterval failover_interval
   failedReplicaPercent percent_value
  exit
 exit
```

注：

- **pods** *pod_name*：モニター対象のポッドの名前を指定します。たとえば、Cache-pod、rest-ep などです。
- **retryCount** *value*：ポッドが ping に失敗した場合に、ポッドがダウンとしてマークされるまで再試行する再試行カウンタ値を指定します。1 ー 10 の範囲の整数で指定する必要があります。
- **retryInterval** *interval_value*：ポッドが ping に成功した場合の再試行間隔をミリ秒単位で指定します。200 ～ 10000 の範囲の整数で指定する必要があります。
- **retryFailOverInterval** *failover_interval*：ポッドが ping に失敗した場合の再試行間隔をミリ秒単位で指定します。200 ～ 10000 の範囲の整数で指定する必要があります。
- **failedReplicaPercent** *percent_value*：ラック間冗長性フェールオーバーがトリガーされるまでの、失敗したレプリカのパーセント値を指定します。10 ～ 100 の範囲の整数で指定する必要があります。

設定例

次に、設定例を示します。

```
geomonitor podmonitor pods cache-pod
 retryCount 3
 retryInterval 5
 retryFailOverInterval 1
 failedReplicaPercent 40
exit
```

リモート クラスタ モニタリング

リモート クラスタ モニタリングは、ロール（リモートラックが **STANDBY_ERROR** 状態の場合はセルフプライマリになる）を自動修正して、トラフィックのトラフィックフローが中断されないようにします。ただし、このロールの自動修正は、特定のロールに対してのみ実行されます。

この機能を設定するには、次の設定例を使用します。

```
config
   geomonitor
      remoteclustermonitor
         retryCount value
         retryInterval interval_value
         end
```

注：

- **retryCount** *value*：現在のラックを **PRIMARY** にする前の再試行回数を指定します。1 ー 10 の範囲の整数で指定する必要があります。デフォルト値は 3 です。
- **retryInterval** *interval_value*：リモートラックのステータスが取得されるまでの再試行間隔をミリ秒単位で指定します。200 ～ 50000 の範囲の整数で指定する必要があります。デフォルト値は 3000 です。

設定例

次に設定例を示します。

```
geomonitor remoteclustermonitor
retryCount 3
retryInterval 3000
```

トラフィック モニタリング

次のコマンドを使用して、トラフィックをモニターします。

```
config
 geomonitor
  trafficMonitor
   thresholdCount value
   thresholdInterval interval_value
  exit
 exit
```

注：

- **thresholdCount** *value*：スタンバイインスタンスの受信コール数を指定します。0 ー 10000 の範囲の整数で指定する必要があります。デフォルト値は 0 です。UDP プロキシと REST-EP の両方をカウンタ値と見なす必要があります。

- **thresholdInterval** *interval_value*：しきい値カウント値に達するまでの最長時間をミリ秒単位で指定します。100 ー 10000 の範囲の整数で指定する必要があります。デフォルト値は 3000 です。

設定例

次に設定例を示します。

```
geomonitor trafficmonitor
thresholdCount 3
thresholdInterval 3000
```

BFD モニタリング

Bidirectional Forwarding Detection（BFD）プロトコルは、迅速なネットワーク障害検出のために BGP とともに使用されます。クラスタ（NF）との BGP ピアリングの間に接続障害が発生するたびに、フェールオーバーがトリガーされ、トラフィック障害の影響が最小限に抑えられます。

```
config
  router bgp as
    bfd interval interval min_rx min_rx multiplier multiplier
    loopbackPort loopbackPort loopbackBFDPort loopbackBFDPort
  interface interface_id (BGP on non-bonded interface <-- loopbackEth)
    bondingInterface bondingInterface (leaf6-nic)
    bondingInterface bondingInterface (leaf6-nic)
    neighbor neighbor_ip_address remote-as remote_as fail-over fail_over_type
  exit
  interface interface_id (BGP on non-bonded interface <-- loopbackEth)
    bondingInterface bondingInterface (leaf7-nic)
    bondingInterface bondingInterface (leaf7-nic)
    neighbor bondingInterface remote-as remote_as fail-over fail_over_type
  exit
  policy-name policy_name
   as-path-set as_path_set
   gateWay gateWay_address
   interface interface_id_source
   ip-prefix ip_prefix_value
   isStaticRoute false | true
   mask-range mask_range
   modifySourceIp false | true
  exit
exit
```

注：

- **bgp** *as*：自律システム（AS）パスセットを指定します。
- **bfd**：BFD 構成を指定します。
  - **interval** *interval* ：BFD 間隔をミリ秒単位で指定します。

- **min_rx** *min_rx*：BFD の最小 RX をミリ秒単位で指定します。
- **multiplier** *multiplier*：BFD 間隔の乗数を指定します。

- **interface** *interface_id*：BGP ローカルインターフェイスを指定します。
  - **bondingInterface** *bondingInterface*：リンクされたボンディング インターフェイスを指定します。
  - **neighbor** *neighbor_ip_address*：ネイバーの IP アドレスを指定します。
    - **fail-over** *fail_over_type*：フェールオーバータイプを指定します。
    - **remote-as** *remote_as*：BGP ネイバーの自律システム（AS）番号を指定します。

- **learnDefaultRoute**：デフォルトルートを学習し、カーネルスペースに追加します
- **loopbackBFDPort** *loopbackBFDPort*：BFD ローカルポートを指定します。
- **loopbackPort** *loopbackPort*：BGP ローカルポートを指定します。
- **policy-name** *policy_name*：ポリシー名を指定します。
  - **as-path-set** *as_path_set*：自律システム（AS）パスセットを指定します。
  - **gateWay** *gateWay_address*：ゲートウェイアドレスを指定します。
  - **interface** *interface_id_source*：送信元 IP として設定するインターフェイスを指定します。
  - **ip-prefix** *ip_prefix_value*：IP プレフィックス値を指定します。
  - **isStaticRoute** *false | true*：カーネルスペースでスタティックルートを追加するかどうかを指定します。デフォルト値は false です。
  - **mask-range** *mask_range*：マスク範囲を指定します。
  - **modifySourceIp** *false | true*：着信ルートの送信元 IP を変更します。デフォルト値は false です。

    true：このオプションは、UDP 関連でない VIP に使用されます。特定のインターフェイスの送信元 IP は、cnSGW-C からパケットを送信するときに送信元 IP として使用されます。

    false：このオプションは、UDP 関連のすべての VIP に使用されます。VIP は、cnSGW-C からパケットを送信するときに送信元 IP として使用されます。

例

設定例は次のとおりです。

```
router bgp 65000
  bfd interval 250000 min_rx 250000 multiplier 3
  loopbackPort 179 loopbackBFDPort 3784
```

```
interface ens160 (BGP on non-bonded interface <-- loopbackEth)
  bondingInterface enp216s0f0 (leaf6-nic)
  bondingInterface enp216s0f1 (leaf6-nic)
  neighbor leaf6-ip remote-as 60000 fail-over bfd
exit
interface ens192 (BGP on non-bonded interface <-- loopbackEth)
  bondingInterface enp94s0f1 (leaf7-nic)
  bondingInterface enp94s0f0 (leaf7-nic)
  neighbor leaf7-ip remote-as 60000 fail-over bfd
exit
policy-name allow-all ip-prefix 209.165.201.30/0 mask-range 0...32
exit
```

BFD を使用した BGP ルータ構成

```
show running-config router
router bgp 65142
 learnDefaultRoute false
 bfd interval 250000 min_rx 250000 multiplier 3
 interface enp94s0f0.3921
  bondingInterface enp216s0f0
  bondingInterface enp94s0f0
  neighbor 209.165.201.24 remote-as 65141 fail-over bfd
 exit
 interface enp94s0f1.3922
  bondingInterface enp216s0f1
  bondingInterface enp94s0f1
  neighbor 209.165.202.24 remote-as 65141 fail-over bfd
```

ネイバーの BFD ステータスを表示

```
show bfd-neigbor
status-details

----- bgpspeaker-pod-1----

Peer             Status

209.165.202.142  STATE_DOWN
----- bgpspeaker-pod-2----

Peer             Status

209.165.202.142  STATE_UP
```

```
policy-name allow-n11 ip-prefix 209.165.200.225/54 mask-range 25..32 interface bd1.n11.2271
 modifySourceIp true isStaticRoute true gateWay 209.165.201.14
```

上記の例では、*modifySourceIp* は true に設定されています。

- AMF サブネット：209.165.200.225/54

  N11 Svc ボンド物理インターフェイス：bd1.n11.2271（IP アドレス：209.165.201.23）

  N11 Svc ボンド VxLAN Anycast GW：209.165.201.14

  N11 VIP アドレス：209.165.201.7

- cnSGW-C アウトバウンドパケット（送信元 IP は 209.165.201.23 になります）

  cnSGW-C へのインバウンドパケット（宛先 IP は 209.165.201.7 になります）

```
policy-name allow-n4-1 ip-prefix 209.165.201.17/41 mask-range 24..32 interface bd2.n4.2274
 gateWay 209.165.201.17
```

上記の例では、*modifySourceIp* は false（デフォルト）に設定されています。

- UPF N4 インターフェイス IP：209.165.201.17/41

  N4 Svc ボンド物理インターフェイス：bd2.n4.2274（IP アドレス：209.165.201.23）

  N4 Svc ボンド VxLAN Anycast GW：209.165.201.17

  N4 VIP アドレス：209.165.201.14

- cnSGW-C アウトバウンドパケット（送信元 IP は 209.165.201.14 になります）

  cnSGW-C へのインバウンドパケット（宛先 IP は 209.165.201.14 になります）

CDL GR の展開

デフォルトでは、CDL は db-ep の 2 つのレプリカ、1 つのスロットマップ（マップごとに 2 つのレプリカ）、および 1 つのインデックスマップ（マップごとに 2 つのレプリカ）で展開されます。

（注） YANG の CDL コンテナを設定することを推奨します。

CDL GR の前提条件

CDL GR を展開する前に、ユーザーは次を構成する必要があります。

- CDL セッションデータベースを設定し、基本設定を定義します。
- CDL 用の Kafka を設定します。
- CDL 用の Zookeeper を設定します。

CDL インスタンス認識とレプリケーション

CDL では、既存の GR 関連パラメータとともに、すべてのラックで機能フラグを使用して GR インスタンス認識を有効にする必要があります。また、この機能をすべてのラックで機能させるには、システム ID からスライス名へのマッピングも提供する必要があります。

CDL には Geo レプリケーション（GR）フェールオーバー通知も用意されていているため、セッションデータのタイマーの期限切れ、および現在アクティブなラックへの一括通知を通知できます。CDL は GR フェールオーバー通知のために、App-Infra を介してボーダー ゲートウェイ プロトコル（BGP）を使用します。

CDL は両方の GR ラックのキー値を登録します。登録したキー値に変更があると、App-Infra は CDL に通知を送信します。キー値は、CDL システム ID または GR インスタンスの状態を示します。GR インスタンスは、 キーの CDL システム ID または GR インスタンス ID を使用して CDL スライスにマッピングされます。

システム ID は両方のラックで必須です。NF 構成の GR インスタンス ID が CDL システム ID と一致している必要があります。

CDLには、インスタンス固有のデータスライスがあります。また、ユーザーは起動時にインスタンス固有のスライス情報を設定できます。

- CDLは、有効期限切れ時またはアクティブなスライスからの一括通知要求時にデータを通知します。
- CDL は、app-infra memory-cache からの通知に基づいてアクティブインスタンスを決定します。
- CDL スライスは、異なる種類のデータを保存する CDL インスタンス内のパーティションです。この場合、NF はデータの別のインスタンスを保存します。

（注） CDL スライス名は、GR で設定されているスライス名と一致する必要があります。

CDL インスタンス認識の構成

CDL インスタンス認識を構成するには、次のコマンドを使用します。

```
config
cdl
 datastore datastore_session_name
   features
    instance-aware-notification
     enable [ true | false ]
     system-id system_id
      slice-names slice_names
   end
```

注：

- **datastore** *datastore_session_name*：データストア名を指定します。
- **enable [true | false]**：スライスの GR インスタンス状態チェックを有効にします。
- **system-id** *system_id*：システム ID からスライス名へのマッピング。
- **slice-names** *slice_names*：システム ID に関連付けられているスライス名のリストを指定します。CDL スライス名は、GR で設定されているスライス名と一致する必要があります。

例

次に、設定例を示します。

```
cdl datastore session
 features instance-aware-notification enable true
 features instance-aware-notification system-id 1
  slice-names [ sgw1 smf1 ]
 exit
 features instance-aware-notification system-id 2
```

```
 slice-names [ sgw2 smf2 ]
end
```

CDL レプリケーションの設定

ここでは、CDL レプリケーションの設定について説明します。

1. Geo HA 関連の構成パラメータを使用せずに、Rack-1 CDL HA システムを設定します。
   1. 設定でシステム ID を 1 に設定します。
   2. 要件に従って、スロットマップ/レプリカとインデックスマップ/レプリカと Kafka レプリカを設定します。

   以下に設定例を示します。

   ```
   cdl system-id 1
   cdl node-type session
   cdl datastore session
   endpoint replica replica_id
     slot map 4
     slot replica 2
     index map 1
     index replica 2
   cdl kafka replica 2
   ```

1. Rack-2 から Rack-1 への通信用に、Rack-1 で外部 IP を設定します。
   1. Rack-1 で Geo レプリケーションを有効にし、リモートラックを Rack-1 の 2 として設定します。

      ```
      cdl enable-geo-replication true
      ```
   2. Rack-2 からアクセスできるように CDL エンドポイントの外部 IP を設定します。

      ```
      cdl datastore session endpoint external-ip site-1_external_ip
      ```
   3. すべての Kafka レプリカの外部 IP とポートを設定します。

      したがって、Kafka 用に 2 つのレプリカ（デフォルト）が設定されている場合、ユーザーは異なる <ip>+ <port> ペアでサポートされます。

      ```
      cdl kafka external-ip site-1_external_ip port1 cdl kafka external-ip
      site-1_external_ip port2
      ```

2. Rack-2 のリモートラック情報を追加します。
   - Rack-2 のリモートラック cdl-ep の設定：

     ```
     cdl remote-site 1 db-endpoint host site-1_cdl_ep_ip
     cdl remote-site 1 db-endpoint port site-1_cdl_ep_port
     ```

     （ポートの例：8882）
   - Rack-2 のリモートラック Kafka 設定：

     ```
     cdl remote-site 1 kafka-server site-1_kafka1_ip site-1_kafka1_port
     ```

```
cdl remote-site 1 kafka-server site-1_kafka2_ip site-1_kafka2_port
```

- セッションデータストアコンフィギュレーションをリモートRack-2コンフィギュレーションに誘導します。

  ```
  cdl datastore session geo-remote-site 1
  ```

- （任意）SSL証明書を設定して、Rack-1のリモートラックとのセキュアな接続を確立します。証明書はすべて、複数行の生のテキスト形式です。証明書が有効でない場合、サーバーの接続は非セキュアな接続のままです。

  cdl ssl-config certs *site-2_external_ip* **ssl-key** *<ssl_key>*

  cdl ssl-config certs *site-2_external_ip* **ssl-crt** *<ssl_crt>*

3. Rack-2 でGR設定をコミットします。
   - 設定をコミットし、ポッドをRack-2に展開します。
   - すべてのポッドが実行状態であることを確認します。
   - 両方のラックを展開したら、両方のラックのミラーメーカーポッドが実行中で、準備完了状態であることを確認します。

例

HA：

```
cdl node-type db-ims

cdl datastore session
 endpoint replica 2
 index map    1
 index write-factor 1
 slot replica 2
 slot map     4
 slot write-factor 1
exit

k8 label cdl-layer key smi.cisco.com/node-type value smf-ims-session
```

Rack-1：

```
cdl system-id          1
cdl node-type          session
cdl enable-geo-replication true
cdl zookeeper replica 1

cdl remote-site 2
 db-endpoint host 209.165.201.21 >> Rack-2 external CDL IP
 db-endpoint port 8882
 kafka-server 209.165.201.21 10092 >> Rack-2 external CDL IP
 exit
exit

cdl label-config session
 endpoint key smi.cisco.com/node-type1
 endpoint value smf-cdl
 slot map 1
```

```
  key   smi.cisco.com/node-type1
  value smf-cdl
 exit
 index map 1
  key   smi.cisco.com/node-type1
  value smf-cdl
 exit
exit
cdl logging default-log-level debug

cdl datastore session
 label-config    session
 geo-remote-site [ 2 ]
 slice-names     [ 1 2 ]
 endpoint cpu-request 100
 endpoint replica 2
 endpoint external-ip 209.165.201.25 >> Rack-1 external CDL IP
 endpoint external-port 8882
 index cpu-request 100
 index replica 2
 index map   1
 slot cpu-request 100
 slot replica 2
 slot map    1
exit

cdl kafka replica 1
cdl kafka label-config key smi.cisco.com/node-type1
cdl kafka label-config value smf-cdl
cdl kafka external-ip 209.165.201.25 10092 >> Rack-1 external CDL IP
```

Rack-2：

```
cdl system-id          2
cdl node-type          session
cdl enable-geo-replication true
cdl zookeeper replica 1

cdl remote-site 1
 db-endpoint host 209.165.201.25 >> Rack-1 external CDL IP
 db-endpoint port 8882
 kafka-server 209.165.201.25 10092 >> Rack-1 external CDL IP
 exit
exit

cdl label-config session
 endpoint key smi.cisco.com/node-type12
 endpoint value smf-cdl
 slot map 1
  key   smi.cisco.com/node-type12
  value smf-cdl
 exit
 index map 1
  key   smi.cisco.com/node-type12
  value smf-cdl
 exit
exit

cdl datastore session
 label-config    session
 geo-remote-site [ 1 ]
 slice-names     [ 1 2 ]
 endpoint cpu-request 100
 endpoint replica 2
 endpoint external-ip 209.165.201.21 >> Rack-2 external CDL IP
```

```
 endpoint external-port 8882
 index cpu-request 100
 index replica 2
 index map   1
 slot cpu-request 100
 slot replica 2
 slot map    1
exit

cdl kafka replica 1
cdl kafka label-config key smi.cisco.com/node-type12
cdl kafka label-config value smf-cdl
cdl kafka external-ip 209.165.201.21 10092 >> Rack-2 external CDL IP
```

合法的傍受

合法的傍受（LI）機能を使用すると、法執行機関（LEA）がサブスクライバの通信を傍受できます。LI機能を使用すると、ネットワークオペレータは、対象のモバイルユーザーの制御メッセージとデータメッセージを傍受できるようになります。このサポートを呼び出すため、LEAはネットワークオペレータに特定のモバイルユーザーの傍受を開始するように要求します。法的な承認がこの要求をサポートします。

1. 合法的傍受（LI）タップは、すべてのラックで設定/有効にする必要があります。1 つのラックでLI設定に失敗した場合、特定のサブスクライバのタップをすべてのラックで有効にするようにLEA を再設定する必要があります。

（注） LI タップ設定はラック間で同期されません。

そのため、LI タップ設定はすべてのラックで必須となります。

LI タップ設定の詳細については、シスコのテクニカル担当者にお問い合わせください。

2. GR インスタンス認識は、合法的傍受 src-address にのみ適用されます。

   例：

   ```
   lawful-intercept instance 1 src-addr 209.165.200.225
   ```

   または

   ```
   lawful-intercept
     instance 1
       src-addr 209.165.200.225
   ```

3. `show` コマンドはインスタンスを認識しません。特定のクラスタで設定されているすべてのタップを表示します。

   LI `show` コマンドの詳細については、シスコのテクニカル担当者にお問い合わせください。

4. すべてのGRインスタンスがクラスタでスタンバイ状態にあり、アクティブなLIタップが失敗し、CLI メッセージ「ラックがスタンバイモードにあるため、アクティブなタップは許可されません。`camp on` を試行してください」が表示された場合、同じサブスクライバで camp-on タップを設定します。

RADIUS 設定

NAS-IP と NAS-Identifier はインスタンス対応です。プロファイル RADIUS 構成では、instance-id ごとに異なる NAS-IP と NAS-Identifier を構成できます。既存の非インスタンス NAS-IP および NAS-Identifier 構成が、ラックのローカルインスタンスのデフォルト nas-ip およびデフォルト nas-id として使用されます。

例

次の構成例を参照してください。

```
profile radius
 attribute
  instance 1
   nas-ip 209.165.200.225  --> Instance-1 specific NAS-IP, used for common AUTH & ACCT
   nas-identifier  smf1    --> Instance-1 specific NAS-Identifier, used for common AUTH
 & ACCT
  exit
  instance 2
   nas-ip 209.165.200.230  --> Instance-2 specific NAS-IP, used for common AUTH & ACCT
   nas-identifier  smf2    --> Instance-2 specific NAS-Identifier, used for common AUTH
 & ACCT
  exit
 exit
 accounting
  attribute
   instance 1
    nas-ip 209.165.200.225  --> Instance-1 specific NAS-IP, used for common ACCT
    nas-identifier  smf1    --> Instance-1 specific NAS-Identifier , used for common
ACCT
   exit
   instance 2
    nas-ip 209.165.200.230  --> Instance-2 specific NAS-IP, used for common ACCT
    nas-identifier  smf2    --> Instance-2 specific NAS-Identifier , used for common
ACCT
   exit
  exit
 exit
 server-group g1
  attribute
   instance 1
    nas-ip 209.165.200.225  --> Instance-1 specific NAS-IP, used for server-group <g1>
AUTH & ACCT
    nas-identifier  smf1    --> Instance-1 specific NAS-ID, used for server-group <g1>
Auth &Acct
   exit
   instance 2
    nas-ip 209.165.200.230  --> Instance-2 specific NAS-IP, used for server-group <g1>
AUTH & ACCT
    nas-identifier  smf2    --> Instance-2 specific NAS-ID,used for server-group
<g1>AUTH&ACCT
   exit
  exit
  accounting
   attribute
    instance 1
     nas-ip 209.165.200.225 --> Instance-1 specific NAS-IP, used for server-group <g1>
ACCT
     nas-identifier  smf1   --> Instance-1 specific NAS-ID, used for server-group <g1>
ACCT
    exit
```

```
    instance 2
     nas-ip 209.165.200.230 --> Instance-2 specific NAS-IP, used for server-group <g1>
ACCT
     nas-identifier  smf2   --> Instance-2 specific NAS-ID, used for server-group <g1>
ACCT
    exit
   exit
  exit
 exit
exit
```

endpoint ポッド構成が特定のインスタンスの下に移動するため、Radius Disconnect-Request VIP もインスタンス対応になります。

```
instance instance-id 1
 endpoint radius
  replicas 1
  interface coa-nas
   vip-ip 209.165.202.130 vip-port 3799  --> Instance-1 specific Radius-Disconnect-Msg-VIP
 & PORT
  exit
 exit
exit
instance instance-id 2
 endpoint radius
  replicas 1
  interface coa-nas
   vip-ip 209.165.202.129 vip-port 3799  --> Instance-2 specific Radius-Disconnect-Msg-VIP
 & PORT
  exit
 exit
exit
```

GR ペアのソフトウェアアップデート

`config commit` を基準として考慮してください。同じチェックリストは、他のアップグレードシナリオにも適用できます。

チェックリスト

（注） 両方のラック（ラック 1 とラック 2）で `cluster sync` を同時に実行しないでください。ラック 1 のアップグレードを続行する前に、ラック 1 で手動スイッチオーバーをトリガーしてください。

- 両方のラックで同時に `config commit` を実行しないでください。各ラックで個別に `config commit` を実行してください。
- ラック 1 で `config commit` を実行する前に、ラック 1 で CLI ベースのスイッチオーバーを開始し、ラック 2 に両方のインスタンス（instance-id 1 と instance-id 2）のプライマリ所有権があることを確認してください。
- ラック 1 で `config commit` を実行します。`config commit` が成功し、ポッドが再起動し、実行状態に戻って最新の helm チャートを取得するのを待ちます（該当する場合）。

- ラック 1 のロールをプライマリに戻します（両方のラックのロールを切り替え/リセットします）。
- ラック 1（プライマリ）とラック 2（スタンバイ）の使用可能なロールが、予期されるステータスになっていることを確認します。
- ラック 2 についてもこのチェックリストを繰り返します。

ソフトウェア アップグレード

GR が有効な場合のラック 1 のアップグレード：

1. ラック 1 の両方のインスタンスで使用可能なロールが PRIMARY/STANDBY であることを確認します。

```
show role instance-id 1
result "PRIMARY"

show role instance-id 2
result "STANDBY"
```

2. フェールバック間隔を 0 秒にして、ラック 1 の両方のインスタンスのロールの STANDBY への切り替えを開始します。このステップにより、ロールが PRIMARY/STANDBY から STANDBY_ERROR/STANDBY_ERROR に移行します。

```
geo switch-role instance-id 1 role standby [failback-interval 0]
geo switch-role instance-id 2 role standby [failback-interval 0]
```

（注）
- 両方のラック間のハートビートが正常である必要があります。

3. ラック 1 の両方のインスタンスの使用可能なロールが、STANDBY_ERROR に変わったことを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "STANDBY_ERROR"
```

4. ラック 2 の両方のインスタンスの使用可能なロールが、PRIMARY に変わったことを確認します。

```
show role instance-id 1
result "PRIMARY"

show role instance-id 2
result "PRIMARY"
```

5. ラック 1 の要件に従って、システムモードの shutdown/running を使用して、ローリングアップグレードまたは非グレースフルアップグレードを実行します。レプリケーションが完了するまで、GR のスイッチオーバーと SMF のシャットダウンの間に 5 分の間隔を空けてください。

6. アップグレード手順の完了後に次の手順を実行します。ラック 1 で正常性チェックを実行して、POD が起動し、ラック 1 が正常であることを確認します。

7. ラック 1 の両方のインスタンスの使用可能なロールが、STANDBY_ERROR モードのままであることを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "STANDBY_ERROR"
```

8. ラック 1 の両方のインスタンスのロールの STANDBY へのリセットを開始します。このステップにより、ロールが STANDBY_ERROR/STANDBY_ERROR から STANDBY/STANDBY に移行します。

```
geo reset-role instance-id 1 role standby
geo reset-role instance-id 2 role standby
```

9. ラック 1 の両方のインスタンスのロールが STANDBY に変わったことを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "STANDBY"
```

10. ラック 2 の instance-id 1 のロールの STANDBY への切り替えを開始します。この手順により、ラック 2 の使用可能なロールが PRIMARY/PRIMARY から STANDBY_ERROR/PRIMARY に、ラック 1 の使用可能なロールが STANDBY/STANDBY から PRIMARY/STANDBY に移行します。

```
geo switch-role instance-id 1 role standby [failback-interval 0]
```

11. ラック 2 のインスタンスで使用可能なロールが STANDBY_ERROR/PRIMARY であることを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "PRIMARY"
```

12. ラック 1 の両方のインスタンスで使用可能なロールが PRIMARY/STANDBY であることを確認します。

```
show role instance-id 1
 result "PRIMARY"

show role instance-id 2
 result "STANDBY"
```

13. ラック 2 の instance-id 1 のロールの STANDBY へのリセットを開始します。このステップにより、ラック 2 のロールが STANDBY_ERROR/PRIMARY から STANDBY/PRIMARY に移行します。

```
geo reset-role instance-id 1 role standby
```

14. ラック 2 の両方のインスタンスで使用可能なロールが STANDBY/PRIMARY であることを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
 result "PRIMARY"
```

GR が有効な場合のラック 2 のアップグレード：

1. ラック 2 の両方のインスタンスで使用可能なロールが STANDBY/PRIMARY であることを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "PRIMARY"
```

2. フェールバック間隔を 0 秒にして、ラック 2 の両方のインスタンスのロールの STANDBY への切り替えを開始します。このステップにより、ロールが STANDBY/PRIMARY から STANDBY_ERROR/STANDBY_ERROR に移行します。

```
geo switch-role instance-id 1 role standby [failback-interval 0]
geo switch-role instance-id 2 role standby [failback-interval 0]
```

3. ラック 2 の両方のインスタンスの使用可能なロールが、STANDBY_ERROR に変わることを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "STANDBY_ERROR"
```

4. ラック 1 の両方のインスタンスの使用可能なロールが、PRIMARY に変わることを確認します。

```
show role instance-id 1
 result "PRIMARY"

show role instance-id 2
 result "PRIMARY"
```

5. ラック 2 の要件に従って、システムモードの shutdown/running を使用して、ローリングアップグレードまたは非グレースフルアップグレードを実行します。

6. アップグレード手順の完了後に次の手順を実行します。ラック 2 で正常性チェックを実行して、POD が起動し、ラック 2 が正常であることを確認します。

7. ラック 2 の両方のインスタンスの使用可能なロールが、STANDBY_ERROR のままであることを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "STANDBY_ERROR"
```

8. ラック 2 の両方のインスタンスのロールの STANDBY へのリセットを開始します。このステップにより、ロールが STANDBY_ERROR/STANDBY_ERROR から STANDBY/STANDBY に移行します。

```
geo reset-role instance-id 1 role standby
geo reset-role instance-id 2 role standby
```

9. ラック 2 の両方のインスタンスの使用可能なロールが、STANDBY に変わることを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "STANDBY"
```

10. ラック 1 の instance-id 2 のロールの STANDBY への切り替えを開始します。この手順により、ラック 1 の使用可能なロールが PRIMARY/PRIMARY から PRIMARY/STANDBY_ERROR に、ラック 1 の使用可能なロールが STANDBY/STANDBY から STANDBY/PRIMARY に移行します。

```
geo switch-role instance-id 2 role standby [failback-interval 0]
```

11. ラック 1 の両方のインスタンスで使用可能なロールが PRIMARY/STANDBY_ERROR であることを確認します。

```
show role instance-id 1
result "PRIMARY"

show role instance-id 2
result "STANDBY_ERROR"
```

12. ラック 2 の両方のインスタンスで使用可能なロールが STANDBY/PRIMARY であることを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "PRIMARY"
```

13. ラック 1 の instance-id 2 のロールの STANDBY へのリセットを開始します。このステップにより、ラック 1 のロールが PRIMARY/STANDBY_ERROR から PRIMARY/STANDBY に移行します。

```
geo reset-role instance-id 2 role standby
```

14. ラック 1 の両方のインスタンスで使用可能なロールが PRIMARY/STANDBY であることを確認します。

```
show role instance-id 1
result "PRIMARY"

show role instance-id 2
result "STANDBY"
```

GR CLI

ここでは、GR CLI ベースのコマンドについて説明します。

Geo ロールの切り替え

GR ロールを切り替えるには、プライマリラックでコマンドを開始し（たとえば、**PRIMARY** ロールから **STANDBY** のみ）、次のコマンドを使用します。

geo switch-role { role *primary | standby* **instance-id** *gr_instanceId* **[failback-interval** *failback_interval* **] }**

注：

- **role** *role*：特定のラックに新しいロールを指定します。

  ロールは、*primary* または *standby* です。特定の GR インスタンス ID のプライマリロールから手動スイッチオーバーをトリガーする必要があります。

- **instance-id** *gr_instanceId*：GR インスタンス ID を指定します

- **failback-interval** は、リリース間のアップグレードの下位互換性を提供するオプションのコマンドです。**failback-interval** の推奨値は 0 です。

重要 **geo switch-role** コマンドは、特定のインスタンス ID について、あるラックから別のラックへの手動フェールオーバーをトリガーします。フェールオーバーをトリガーするラックが、**PRIMARY** ロールから **STANDBY_ERROR** ロールに変更されます。その間、フェールオーバーをトリガーするラックは、別のラックにフェールオーバー（GR トリガー）メッセージを送信します。フェールオーバーメッセージを受信した他のラックは、**STANDBY** ロールから **PRIMARY** ロールに変更されます。

Geo ロールのリセット

GR インスタンスのロールをリセットする（たとえば **STANDBY_ERROR** のロールから **STANDBY**、**PRIMARY** へ）には、次のコマンド例を使用します。

geo reset-role role *role* **instance-id** *gr_instanceId*

注：

- **role** *role*：指定のラックに新しいロールを指定します。

  ロールは **PRIMARY** または **STANDBY** です。

- **instance-id** *gr_instanceId*：GR インスタンス ID を指定します。

重要 **geo reset-role** コマンドは、ローカルラックの特定のインスタンスでロールの変更をトリガーします。リモートラックは、同じコマンドのメッセージを受信しません。特定のインスタンス ID のロールを **STANDBY_ERROR** から **STANDBY** に、および **STANDBY** から **PRIMARY** に変更することのみが可能です。別のロールの変更はできません。

トラブルシューティング

ここでは、さまざまな該当するトラブルシューティングのシナリオについて説明します。

show/clear コマンド

ここでは、問題のデバッグに役立つ show/clear コマンドについて説明します。

clear subscriber

gr-instance 対応サブスクライバをクリアするには、次のコマンドを使用します。

clear subscriber all gr-instance *gr_instanceId*

（注） **gr-instance** はオプションのパラメータです。**gr-instance** が指定されていない場合、`show subscriber all` ではそのラックのローカルインスタンス ID が考慮されます。

例

次に、設定例を示します。

```
clear subscriber all gr-instance 1
result
ClearSubscriber Request submitted
```

show BFD Status

ネイバーの BFD ステータスを表示するには、次のコマンドを使用します。

show bfd-neighbor

例

次に、いくつかの構成例を示します。

```
show bfd-neighbor
status-details

-----example-bgp-ep-1 ----

Peer             Status

 209.165.202.142  STATE_DOWN
-----example-bgp-ep-2 ----

Peer             Status

 209.165.202.142  STATE_DOWN

show bfd-neigbor
status-details

-----bgpspeaker-pod-1 ----

Peer             Status
```

```
 209.165.202.131
-----bgpspeaker-pod-2 ----

Peer             Status

 209.165.202.131   STATE_UP
```

show BGP Global

BGP グローバル構成を表示するには、次のコマンドを使用します。

show bgp-global

例

次に、いくつか設定例を示します。

```
show bgp-global
global-details
-----example-bgp-ep-2 ----
AS:        65000
Router-ID: 209.165.202.149
Listening Port: 179, Addresses: 209.165.202.149
-----example-bgp-ep-1 ----
AS:        65000
Router-ID: 209.165.202.148
Listening Port: 179, Addresses: 209.165.202.148
```

```
show bgp-global
global-details

-----bgpspeaker-pod-2 ----
AS:        65061
Router-ID: 209.165.202.132
Listening Port: 179, Addresses: 209.165.202.132
```

show bgp kernel route

BGP カーネル構成ルートを表示するには、次のコマンドを使用します。

show bgp-kernel-route kernel-route

例

次に、いくつかの構成例を示します。

```
show bgp-kernel-route
kernel-route

-----example-bgp-ep-2 ----

 DestinationIP  SourceIP        Gateway

-----example-bgp-ep-1 ----

 DestinationIP      SourceIP          Gateway
 209.165.202.133    209.165.202.148   209.165.202.142
 209.165.202.134    209.165.202.148   209.165.202.142
```

```
show bgp-kernel-route
kernel-route

-----bgpspeaker-pod-2 ----
```

```
 DestinationIP        SourceIP            Gateway

 209.165.202.135    209.165.202.132    209.165.202.131

-----bgpspeaker-pod-1 ----

 DestinationIP  SourceIP        Gateway
```

show bgp neighbors

BGP ネイバーのステータスを表示するには、次のコマンドを使用します。

show bgp-neighbors neighbor-details
show bgp-neighbors ip *ip_address* **neighbor-details**

例

次に、いくつかの構成例を示します。

```
show bgp-neighbors neighbor-details
-----example-bgp-ep-1 ----
Peer             AS  Up/Down State       |#Received  Accepted
209.165.202.142 60000 00:25:06 Establ      |        3         3
-----example-bgp-ep-2 ----
Peer             AS Up/Down State       |#Received  Accepted
209.165.202.142 60000   never Idle        |        0         0

show bgp-neighbors ip 209.165.202.142 neighbor-details
-----example-bgp-ep-2 ----
BGP neighbor is 209.165.202.142, remote AS 60000
  BGP version 4, remote router ID unknown
  BGP state = ACTIVE
  BGP OutQ = 0, Flops = 0
  Hold time is 0, keepalive interval is 0 seconds
  Configured hold time is 90, keepalive interval is 30 seconds

  Neighbor capabilities:
    multiprotocol:
        ipv4-unicast:   advertised
    route-refresh:      advertised
    extended-nexthop:   advertised
        Local:  nlri: ipv4-unicast, nexthop: ipv6
    4-octet-as: advertised
  Message statistics:
                         Sent       Rcvd
    Opens:                130          0
    Notifications:          0          0
    Updates:                0          0
    Keepalives:             0          0
    Route Refresh:          0          0
    Discarded:              0          0
    Total:                130          0
  Route statistics:
    Advertised:             0
    Received:               0
    Accepted:               0

-----example-bgp-ep-1 ----
BGP neighbor is 209.165.202.142, remote AS 60000
  BGP version 4, remote router ID 209.165.202.136
  BGP state = ESTABLISHED, up for 00:25:20
  BGP OutQ = 0, Flops = 0
```

```
Hold time is 90, keepalive interval is 30 seconds
Configured hold time is 90, keepalive interval is 30 seconds

Neighbor capabilities:
  multiprotocol:
      ipv4-unicast:   advertised and received
  route-refresh:      advertised and received
  extended-nexthop:   advertised
      Local:  nlri: ipv4-unicast, nexthop: ipv6
  4-octet-as: advertised and received
Message statistics:
                      Sent       Rcvd
  Opens:                 1          1
  Notifications:         0          0
  Updates:               1          1
  Keepalives:           51         51
  Route Refresh:         0          0
  Discarded:             0          0
  Total:                53         53
Route statistics:
  Advertised:            0
  Received:              3
  Accepted:              3
```

show bgp route summary

BGP ルートサマリを表示するには、次のコマンドを使用します。

show bgp-route-summary

例

次に、設定例を示します。

```
show bgp-route-summary
route-details
-----example-bgp-ep-1 ----
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 5, Path: 5
-----example-bgp-ep-2 ----
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 2, Path: 2
```

BGP ルートの表示

BGP ルート情報を表示するには、次のコマンドを使用します。

show bgp-routes

例

次に、設定例を示します。

```
show bgp-routes
bgp-route

-----example-bgp-ep-1 ----
   Network              Next Hop            AS_PATH      Age        Attrs
*> 209.165.202.133/24   209.165.202.142       60000      00:25:55   [{Origin: i} {Med:
 0}]
*> 209.165.200.225/32   209.165.202.148                  00:26:00   [{Origin: e}
{LocalPref: 100} {Med: 600}]
```

```
*> 209.165.202.134/24   209.165.202.142        60000      00:25:55   [{Origin: i} {Med:
 0}]
*> 209.165.202.140/24   209.165.202.142        60000      00:25:55   [{Origin: i} {Med:
 0}]
*> 209.165.202.146/32   209.165.202.148                   00:26:00   [{Origin: e}
{LocalPref: 100} {Med: 600}]

-----example-bgp-ep-2 ----
   Network              Next Hop             AS_PATH      Age        Attrs
*> 209.165.200.225/32   209.165.202.149                   00:26:24   [{Origin: e}
{LocalPref: 100} {Med: 600}]
*> 209.165.202.146/32   209.165.202.149                   00:26:24   [{Origin: e}
{LocalPref: 100} {Med: 600}]
```

show endpoint

gr-instance 対応になったエンドポイントを表示するには、次のコマンドを使用します。

show endpoint all grInstance *gr_instanceId*

（注） **grInstance** はオプションのパラメータです。**grInstance** が指定されていない場合、`show subscriber all` ではそのラックのローカルインスタンス ID が考慮されます。

例

次に、設定例を示します。

```
show endpoint all grInstance 1
                                                                                                START
  STOPPED  GR
ENDPOINT                            ADDRESS        TYPE  STATUS   INTERFACE  INTERNAL
TIME      TIME     INSTANCE
-------------------------------------------------------------------------------------------------------
209.165.202.137:2123       209.165.202.137:2123  Udp   Started             false     10
 hours  <none>   1
Gtpu:209.165.202.137:2152  209.165.202.137:2152  Udp   Started  GTPU       false     10
 hours  <none>   1
N4:209.165.202.137:8806    209.165.202.137:8806  Udp   Started  N4         false     10
 hours  <none>   1
S2B-GTP                    209.165.202.138:2124  Udp   Started  s2b        false     10
 hours  <none>   1
S5-GTP                     209.165.202.138:2125  Udp   Started  s5         false     10
 hours  <none>   1
S5S8S2B-GTP                209.165.202.138:2123  Udp   Started  s5s8s2b    false     10
 hours  <none>   1
Sxa:209.165.202.137:8805   209.165.202.137:8805  Udp   Started  SXA        false     10
 hours  <none>   1
n10-1                      209.165.202.139:9010  Rest  Started  N10-1      false     10
 hours  <none>   1
n11-1                      209.165.202.139:9011  Rest  Started  N11-1      false     10
 hours  <none>   1
n40-1                      209.165.202.139:9040  Rest  Started  N40-1      false     10
 hours  <none>   1
n7-1                       209.165.202.139:9007  Rest  Started  N7-1       false     10
 hours  <none>   1
sbi-1                      209.165.202.139:8090  Rest  Started  SBI-1      false     10
 hours  <none>   1
```

ETCD/キャッシュポッドのレプリケーションの表示

etcd および cache-pod データのレプリケーションの詳細を表示するには、次のコマンドを使用します。

show georeplication checksum instance-id *gr_instanceId*

例

次に、設定例を示します。

```
show georeplication checksum instance-id
Value for 'instance-id' (<string>): 1
checksum-details
--              ----    --------
 ID             Type    Checksum
 --             ----    --------
 1              ETCD    1617984439
 IPAM           CACHE   1617984439
 NRFCache       CACHE   1617984439
 NRFSubs        CACHE   1617984439
 IDMGR          CACHE   1617984439
 NRFMgmt        CACHE   1617984439
```

show role

GR インスタンスの現在のロールを表示するには、次のコマンドを使用します。

show role instance-id *gr_instanceId*

（注） ロールで使用可能な値のリストは次のとおりです。

- **プライマリ**
- **STANDBY**
- **FAILOVER_INIT**
- **FAILOVER_COMPLETE**
- **STANDBY_ERROR**
- **FAILBACK_STARTED**

例

次に、いくつか設定例を示します。

```
show role instance-id 1
result
"PRIMARY"

show role instance-id 2
result
"STANDBY"
```

type と address を使用した show ipam dp

インスタンス ID とリモートインスタンスのチャンクを示すフラグを表示するには、次のコマンドを使用します。

show ipam dp { *dp_type* } { *addr_type* }

注：

- **dp** *dp_type*：DP タイプを指定します。
- *addr_type*：IPv4/IPv6 アドレスタイプを指定します。

例

次に、設定例を示します。

```
show ipam dp 209.165.202.145:209.165.202.144  ipv4-addr
================================================================
Flag Indication: S(Static) O(Offline) R(For Remote Instance)
G:N/P Indication: G(GR InstId) N(Native NM InstId) P(Peer NM InstId)
================================================================
StartAddress    EndAddress      AllocContext                    Route              G:N/P
 Utilization Flag
================================================================
209.165.200.240 209.165.200.243 209.165.202.145:209.165.202.144 209.165.200.240/24 1:0/1
 0.00%        R
================================================================
```

show ipam dp

この DP のチャンクを持つすべてのインスタンスを表示するには、次のコマンドを使用します。

show ipam dp *dp_name*

注：

- **dp** *dp_name*：データプレーン割り当て名を指定します。

例

次に、設定例を示します。

```
show ipam dp 209.165.202.145:209.165.202.144
-----------------------------------------------------------
Ipv4Addr [Total/Used/Utilization] = 257 / 1 / 0.39%
Ipv6Addr [Total/Used/Utilization] = 0 / 0 / 0.00%
Ipv6Prefix [Total/Used/Utilization] = 2048 / 0 / 0.00%
Instance ID = 1
-----------------------------------------------------------
```

show ipam pool

プールが構成されているインスタンス ID 情報を表示するには、次のコマンドを使用します。

show ipam pool *pool_name*

注：

- **pool** *pool_name*：プール名を指定します。

例

次に、いくつかの構成例を示します。

```
show ipam pool

=============================================================================================
PoolName                     Ipv4Utilization  Ipv6AddrUtilization  Ipv6PrefixUtilization
=============================================================================================
poolv6DNN2                     0.00%            0.00%                0.00%
poolv6                         0.00%            0.00%                0.00%
poolv4vDNN                     0.00%            0.00%                0.00%
poolv4DNN2                     0.00%            0.00%                0.00%
poolv4                         0.00%            0.00%                0.00%
poolv6vDNN                     0.00%            0.00%                0.00%
poolv4DNN3                     -                -                    -
=============================================================================================

show ipam pool poolv4DNN3
-----------------------------------------------------------
Ipv4Addr [Total/Used/Utilization] = 2814 / 0 / -
Ipv6Addr [Total/Used/Utilization] = 0 / 0 / -
Ipv6Prefix [Total/Used/Utilization] = 65536 / 0 / -
Instance ID = 1
isStatic = true
-----------------------------------------------------------

show ipam pool poolv4
-----------------------------------------------------------
 Ipv4Addr   [Total/Used/Utilization] = 2814 / 0 / 0.00%
 Ipv6Addr   [Total/Used/Utilization] = 0 / 0 / 0.00%
 Ipv6Prefix [Total/Used/Utilization] = 0 / 0 / 0.00%
 Instance ID                         = 1
-----------------------------------------------------------
```

show nrf discovery-info discovery-filter

GR インスタンス ID 情報を表示して、検出フィルタ情報が属する GR インスタンスを判断するには、次のコマンドを使用します。

show nrf discovery-info *nf_type* **discovery-filter**

例

次に、設定例を示します。

```
=================================================
-------------------------------------------------------
Discovery Filter: dnn=intershat;
Expiry Time: 1580146356
GR Instance ID: 1
-------------------------------------------------------
=========================================================================
```

show nrf discovery-info

GR インスタンス ID 情報を表示して、検出情報が属する GR インスタンスを判別するには、次のコマンドを使用します。

show nrf discovery-info

例

次に、設定例を示します。

```
show nrf discovery-info
=====================================================
------Discovered NFs:-------
    NF Type: AMF
    Number of Discovery Filters: 15
    Number of NF Profiles: 15
    GR Instance ID: 1
------Discovered NFs:-------
    NF Type: UDM
    Number of Discovery Filters: 1
    Number of NF Profiles: 3
    GR Instance ID: 2
=====================================================
```

show nrf registration-info

GR インスタンス ID 情報を表示して、登録情報が属する GR インスタンスを特定するには、次のコマンドを使用します。

show nrf registration-info

例

次に、設定例を示します。

```
show nrf registration-info
============================================================================
 NF Status: Not Registered
 Registration Time:
 Active MgmtEP Name:
 Heartbeat Duration: 0
 GR Instance ID: 1
========

show nrf registration-info

============================================================================
 Gr-instance:
 NF Status: Not Registered
 Registration Time:
 Active MgmtEP Name:
 Heartbeat Duration: 0
 Uri:
 Host Type:

============================================================================
 Gr-instance:
 NF Status: Not Registered
 Registration Time:
 Active MgmtEP Name:
 Heartbeat Duration: 0
 Uri:
 Host Type:

============================================================================
```

show nrf subscription-info

サブスクリプション情報が属する GR インスタンスを判別するために GR インスタンス ID 情報を表示するには、次のコマンドを使用します。

show nrf subscription-info

例

次に、設定例を示します。

```
show nrf subscription-info
=====================================================================
NF Instance Id: f9882966-a253-32d1-8b82-c785b34a7cc9
SubscriptionID : subs123459
Actual Validity Time : 2020-01-21 12:39:45 +0000 UTC
Requested Validity Time : 2020-01-21 12:39:45 +0000 UTC
GR Instance ID: 1
=====================================================================
```

show peers

gr-instance 対応になったピアを表示するには、次のコマンドを使用します。

show peers all grInstance *gr_instanceId*

（注） **grInstance** はオプションのパラメータです。**grInstance** が指定されていない場合、`show subscriber all` ではそのラックのローカルインスタンス ID が考慮されます。

例

次に、設定例を示します。

```
show peers all grInstance 1
                                                                  POD              CONNECTED
 ADDITIONAL  INTERFACE  GR
ENDPOINT  LOCAL ADDRESS   PEER ADDRESS     DIRECTION  INSTANCE TYPE TIME     RPC DETAILS
 NAME       INSTANCE
_____________________________________________________________________________________________
<none>  209.165.202.139 209.165.201.22:8001 Outbound rest-ep-0 Rest 10 hours UDM <none>
  n10        1
<none>  209.165.202.139 209.165.201.22:8002 Outbound rest-ep-0 Rest 10 hours AMF <none>
  n11        1
<none>  209.165.202.139 209.165.201.22:8003 Outbound rest-ep-0 Rest 10 hours PCF <none>
  n7         1
<none>  209.165.202.139 209.165.201.22:8004 Outbound rest-ep-0 Rest 10 hours CHF <none>
  n40        1
<none>  209.165.202.139 209.165.201.22:9040 Outbound rest-ep-0 Rest 10 hours CHF <none>
  n40
```

show role

インスタンスのロールを表示するには、次のコマンドを使用します。

show role

例

次に、いくつかの構成例を示します。

```
show role instance-id 2
result "PRIMARY"

show role instance-id 1
result "PRIMARY"
```

show subscriber

gr-instance 対応にするサブスクライバの詳細を表示するには、次のコマンドを使用します。

show subscriber { all | gr-instance *gr_instanceId* **}**

（注） `show subscriber all` は、 ローカル インスタンス サブスクライバの詳細のみを表示します。

gr-instance はオプションのパラメータです。**gr-instance** が指定されていない場合、`show subscriber all` ではそのラックのローカルインスタンス ID が考慮されます。

例

次に、設定例を示します。

```
show subscriber gr-instance 1 all
subscriber-details
{
"subResponses": [
[
""
],
[
""
],
[
"roaming-status:homer",
"supi:imsi-123456789300001",
"gpsi:msisdn-22331010301010",
"psid:1",
"dnn:intershat",
"emergency:false",
"rat:nr",
"access:3gpp access",
"connectivity:5g",
"udm-uecm:209.165.202.150",
"udm-sdm:209.165.202.150",
"auth-status:unauthenticated",
"pcfGroupId:PCF-*",
"policy:2",
"pcf:209.165.202.152",
"upf:209.165.202.154",
"upfEpKey:209.165.202.154:209.165.202.158",
"ipv4-addr:v4pool1/209.165.200.250",
"ipv4-pool:v4pool1",
"ipv4-range:v4pool1/209.165.200.249",
"ipv4-startrange:v4pool1/209.165.200.250",
"id-index:1:0:0:32768",
"id-value:8",
```

```
"chfGroupId:CHF-*",
"chf:209.165.202.151",
"amf:209.165.202.153",
"peerGtpuEpKey:209.165.202.154:209.165.202.155",
"namespace:smf",
"nf-service:smf"
]
]
}
```

モニターサブスクライバ

サブスクライバのメッセージをキャプチャするには（GR インスタンス対応）、次のコマンドを使用します。

（注） 2021.02 以降のリリースでは、**namespace** キーワードは廃止され、**nf-service** キーワードに置き換えられました。

注：

例

次に、設定例を示します。

```
monitor subscriber imsi 123456789 gr-instance 1
supi: imsi-123456789
captureDuration: 300
enableInternalMsg: false
enableTxnLog: false
namespace(deprecated. Use nf-service instead.): none
nf-service: none
gr-instance: 1
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   295  100    98  100   197  10888  21888 --:--:-- --:--:-- --:--:-- 29500
Command: --header Content-type:application/json --request POST --data
{"commandname":"mon_sub","parameters":{"supi":"imsi-123456789","duration":300,
"enableTxnLog":false,"enableInternalMsg":false,"action":"start","namespace":"none",
"nf-service":"none","grInstance":1}} http://oam-pod:8879/commands
Result start mon_sub, fileName
->logs/monsublogs/none.imsi-123456789_TS_2021-04-09T09:59:59.964148895.txt
Starting to tail the monsub messages from file:
logs/monsublogs/none.imsi-123456789_TS_2021-04-09T09:59:59.964148895.txt
Defaulting container name to oam-pod.
Use 'kubectl describe pod/oam-pod-0 -n smf' to see all the containers in this pod.
```

モニタープロトコル

異なるインターフェイスでパケットをキャプチャするには（grインスタンス対応）、次のコマンドを使用します。

monitor protocol { interface *interface_name* **[capture-duration** *duration* **| gr-instance***gr_instance* **| pcacp yes | |] | list [|] }**

注：

- **interface** *interface_name*：PCAP がキャプチャされるインターフェイス名を指定します。この CLI では、1 つの CLI コマンドで複数のインターフェイス名を設定できます。
- **capture-duration** *duration*：pcap がキャプチャされる期間を秒単位で指定します。デフォルトは 300 秒（5 分）です。
- 構成されたインターフェイス名は **show endpoint** 、 CLI コマンドを使用して取得できます。
- **pcap yes**：このオプションを構成すると、PCAP ファイルの生成が有効になります。デフォルトでは、このオプションは無効になっています。
- **list**：プロトコルリストファイルをモニタリングします。

例

次に、設定例を示します。

```
monitor protocol interface sbi gr-instance 1
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100    95  100   125   8636  11363 --:--:-- --:--:-- --:--:-- 20000
Command: --header Content-type:application/json --request POST --data
{"commandname":"mon_pro","parameters":{"interface":"sbi","duration":300,"action":
"start","enable_pcap":false,"grInstance":1}} http://oam-pod:8879/commands
Result start mon_pro, fileName
->logs/monprologs/sessintfname_sbi_at_2021-04-30T05:26:22.712229347.txt
Starting to tail the monpro messages from file:
logs/monprologs/sessintfname_sbi_at_2021-04-30T05:26:22.712229347.txt
Defaulting container name to oam-pod.
Use 'kubectl describe pod/oam-pod-0 -n cn' to see all of the containers in this pod.
```

地理的冗長性 OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

RMA 手順の前提条件

GR 展開の場合、ノードモニターポッドが自動的に開始されます。RRMA 手順中に、ノードがドレインおよび削除され、マルチコンピューティング障害が検出された場合、ノードモニターポッドは自動的にそのラックをシャットダウンします。

RMA（返品許可）の詳細については、『*Ultra Cloud Core*：サブスクライバ マイクロサービス インフラストラクチャ』 の 「*SMI* クラスタの *RMA*」セクションを参照してください。

RMA 手順を開始する前に、次の手順を実行してください。

1. `geo switch-role role` コマンドを使用して両方のインスタンスのロールを他方のラックに切り替え、RMA 対象のラックが両インスタンスともに STANDBY_ERROR ロールになっていることを確認します。
2. ノードモニターポッドを無効にします。
    1. daemonsets のバックアップを取ります。

```
kubectl get daemonsets node-monitor -n cn -o yaml > node-monitor.yaml
```

2. ノードモニターポッドを削除します。

```
kubectl delete daemonsets node-monitor -n cn
```

3. RMA 手順に進みます。詳細については、リンクを参照してください：
4. RMA 手順が完了したら、ノードモニターポッドがすでに生成されているかどうかを確認します。

```
kubectl get pods -n cn -o wide | grep node-monitor
```

ノードモニターポッドが開始していない場合は、再起動します。

```
kubectl create -f node-monitor.yaml
```

（注） `node-monitor.yaml` ファイルは、ステップ 2.a （634 ページ）と同じです。

5. インスタンスのロールを適宜修正します。

（注） 以前の SMI バージョンと現在の SMI バージョンの両方の場合：

- mLOM カードなどのファームウェアを含む RMA の手順中にハードウェアコンポーネントを交換する場合、修理または交換したノードをクラスタに戻す前に HUU（ホスト アップグレードユーティリティ）を実行して、コンポーネントがシステムと互換性があることを確認してからノードをサービスに同期させる必要があります。
- RMA の一環として、クラスタからノードを削除し、それを製造元に返却する場合は、ハードウェアベンダーの指示に従ってデバイス上のすべてのデータを消去する必要があります。

ヘルス チェック

次のセクションでは、GR セットアップの正常性チェックに関する情報を提供します。

- すべての重要なポッドは、ユーザートラフィックを処理できる良好な状態です。

  GR および CDL に関連するポッドが実行状態にあるかどうかを確認するには、次のコマンドを使用します。

```
kubectl get pods -n cn-cn1 -o wide | grep georeplication-pod
kubectl get pods -n cn-cn1 -o wide | grep cdl
kubectl get pods -n cn-cn1 -o wide | grep mirror-maker
```

- キープアライブポッドは正常な状態になり、check-interface/check-port 用に構成されたすべての VIP をモニタリングします。

次のコマンドを使用して、「smi-vips」名前空間のキープアライブポッドが「実行」状態であるかどうかを確認します。

```
kubectl get pods -n smi-vips
```

- CDLに関連するポッドの正常性チェック：CDL DBエンドポイント、スロット、およびインデックスのステータスを確認します。システムID 1と2の両方で、すべてのがSTARTEDまたはONLINE状態になっている必要があります。

```
cdl show status
message params: {cmd:status mode:cli dbName:session sessionIn:{mapId:0 limit:500
key: purgeOnEval:0 filters:[] nextEvalTsStart:0 nextEvalTsEnd:0 allReplicas:false
maxDataSize:4096} sliceName:}
db-endpoint {
    endpoint-site {
        system-id 1
        state STARTED
        total-sessions 4
        site-session-count 2
        total-reconciliation 0
        remote-connection-time 66h37m31.36054781s
        remote-connection-last-failure-time 2021-07-13 11:24:10.233825924 +0000 UTC

        slot-geo-replication-delay 2.025396ms
    }
    endpoint-site {
        system-id 2
        state STARTED
        total-sessions 4
        site-session-count 2
        total-reconciliation 0
        remote-connection-time 66h58m49.83449066s
        remote-connection-last-failure-time 2021-07-13 11:02:51.759971655 +0000 UTC

        slot-geo-replication-delay 1.561816ms
    }
}
slot {
    map {
        map-id 1
        instance {
            system-id 1
            instance-id 1
            records 4
            capacity 2500000
            state ONLINE
            avg-record-size-bytes 1
            up-time 89h38m37.335813523s
            sync-duration 9.298061ms
        }
        instance {
            system-id 1
            instance-id 2
            records 4
            capacity 2500000
            state ONLINE
            avg-record-size-bytes 1
            up-time 89h39m11.1268024s
            sync-duration 8.852556ms
        }
        instance {
            system-id 2
            instance-id 1
```

```
                records 4
                capacity 2500000
                state ONLINE
                avg-record-size-bytes 1
                up-time 89h28m38.274713022s
                sync-duration 8.37766ms
            }
            instance {
                system-id 2
                instance-id 2
                records 4
                capacity 2500000
                state ONLINE
                avg-record-size-bytes 1
                up-time 89h29m37.934345015s
                sync-duration 8.877442ms
            }
        }
    }
    index {
        map {
            map-id 1
            instance {
                system-id 1
                instance-id 1
                records 4
                capacity 60000000
                state ONLINE
                up-time 89h38m16.119032086s
                sync-duration 2.012281769s
                leader false
                geo-replication-delay 10.529821ms
            }
            instance {
                system-id 1
                instance-id 2
                records 4
                capacity 60000000
                state ONLINE
                up-time 89h39m8.47664588s
                sync-duration 2.011171261s
                leader true
                leader-time 89h38m53.761213379s
                geo-replication-delay 10.252683ms
            }
            instance {
                system-id 2
                instance-id 1
                records 4
                capacity 60000000
                state ONLINE
                up-time 89h28m29.5479133s
                sync-duration 2.012101957s
                leader false
                geo-replication-delay 15.974538ms
            }
            instance {
                system-id 2
                instance-id 2
                records 4
                capacity 60000000
                state ONLINE
                up-time 89h29m11.633496562s
                sync-duration 2.011566639s
```

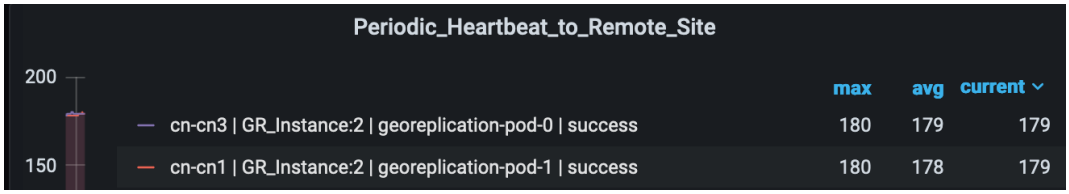
```
            leader true
            leader-time 89h28m51.29928233s
            geo-replication-delay 16.213323ms
        }
    }
}
```

- CDL レプリケーションステータス

  [CDL Replication Stats] Grafana ダッシュボードの [GRPC_Connections_to_RemoteSite] パネルのラック全体で、（各名前空間の）CDL EP セッションポッド間で 4 つの gRPC 接続が確立されているかどうかを確認します。両方のラックで Grafana をチェックします。

- Geo レプリケーション用のラック間の管理ポートステータス。

  [GR Statistics] Grafana ダッシュボードの [Periodic_Heartbeat_to_Remote_Site] パネルで、ラック全体の Geo レプリケーションポッド間のハートビートメッセージを確認します。



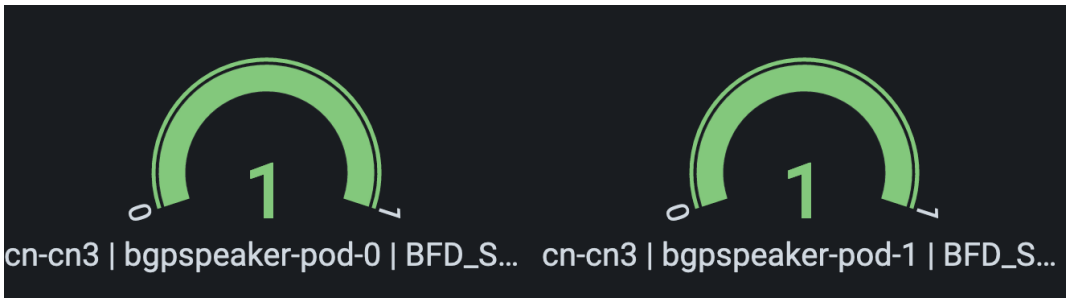
- ラック上の BGP/BFD リンクステータス

  BGP ピアとの隣接関係が確立されているかどうかを、[BGP, BFD Statistics] Grafana ダッシュボードの [BGP Peers] パネルで確立しているかどうかを確認します。

BGP Peers

Time	as_path	namespace	peer_ip	pod	Value
2021-07-16 06:29:18	3333	cn-cn1	192.204.10.6	bgpspeaker-pod-0	1
2021-07-16 06:29:18	3333	cn-cn1	192.204.10.6	bgpspeaker-pod-1	1
2021-07-16 06:29:18	3333	cn-cn3	192.204.18.6	bgpspeaker-pod-0	1
2021-07-16 06:29:18	3333	cn-cn3	192.204.18.6	bgpspeaker-pod-1	1

BFD リンクが接続状態であるかどうかを、[BGP, BFD Statistics] Grafana ダッシュボードの [BFD Link Status] パネルで確認します。



- 各インスタンスのロールが正常な状態である

  各ラックのロールがどの時点においても STANDBY_ERROR 状態でないことを確認します。

  - **アクティブ/スタンバイモデル**：ロールは各ラックで次の状態である必要があります。

    Rack-1：

    ```
    show role instance-id 1
    result "PRIMARY"
    show role instance-id 2
    result "PRIMARY"
    ```

    Rack-2：

    ```
    show role instance-id 1
    result "STANDBY"
    show role instance-id 2
    result "STANDBY"
    ```

  - **アクティブ/アクティブモデル**：ロールは、各ラックで次の状態になっている必要があります。

    Rack-1：

    ```
    show role instance-id 1
    result "PRIMARY"
    show role instance-id 2
    result "STANDBY"
    ```

    Rack-2：

    ```
    show role instance-id 1
    result "STANDBY"
    show role instance-id 2
    result "PRIMARY"
    ```

回復手順

Rack-1 上

1. Rack-1 の両方のインスタンスのロールが STANDBY_ERROR であることを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "STANDBY_ERROR"
```

2. Rack-1 の両方のインスタンスで、STANDBY へのロールのリセットを開始します。このステップにより、ロールが STANDBY_ERROR/STANDBY_ERROR から STANDBY/STANDBY に移行します。

```
geo reset-role instance-id 1 role standby
geo reset-role instance-id 2 role standby
```

3. Rack-1 の両方のインスタンスのロールが STANDBY に移行したことを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "STANDBY"
```

4. フェールバック間隔を 30 秒にして、Rack-2 で instance-id 1 のロールの STANDBY への切り替えを開始します。このステップにより、Rack-2 のロールが PRIMARY/PRIMARY から STANDBY_ERROR/PRIMARY に、Rack-1 のロールが STANDBY/STANDBY から PRIMARY/STANDBY に移行します。

```
geo switch-role instance-id 1 role standby [failback-interval 0]
```

5. Rack-2 の両方のインスタンスのロールが STANDBY_ERROR/PRIMARY であることを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "PRIMARY"
```

6. Rack-1 の両方のインスタンスのロールが PRIMARY/STANDBY であることを確認します。

```
show role instance-id 1
result "PRIMARY"

show role instance-id 2
result "STANDBY"
```

7. Rack-2 の instance-id 1 で、STANDBY へのロールのリセットを開始します。このステップにより、Rack-2 のロールが STANDBY_ERROR/PRIMARY から STANDBY/PRIMARY に移行します。

```
geo reset-role instance-id 1 role standby
```

8. Rack-2 のロールが STANDBY/PRIMARY であることを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "PRIMARY"
```

Rack-2 上

1. Rack-2 の両方のインスタンスのロールが STANDBY_ERROR であることを確認します。

```
show role instance-id 1
result "STANDBY_ERROR"

show role instance-id 2
result "STANDBY_ERROR"
```

2. Rack-2 の両方のインスタンスで、STANDBY へのロールのリセットを開始します。このステップにより、ロールが STANDBY_ERROR/STANDBY_ERROR から STANDBY/STANDBY に移行します。

```
geo reset-role instance-id 1 role standby
geo reset-role instance-id 2 role standby
```

3. Rack-2 の両方のインスタンスのロールが STANDBY に移行することを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "STANDBY"
```

4. Rack-1 の instance-id 2 で、STANDBY へのロールの切り替えを開始します。このステップにより、Rack-1 のロールが PRIMARY/PRIMARY から PRIMARY/STANDBY_ERROR に、Rack-2 のロールが STANDBY/STANDBY から STANDBY/PRIMARY に移行します。

```
geo switch-role instance-id 2 role standby [failback-interval 0]
```

5. Rack-1 のインスタンスのロールが PRIMARY/STANDBY_ERROR モードであることを確認します。

```
show role instance-id 1
result "PRIMARY"

show role instance-id 2
result "STANDBY_ERROR"
```

6. Rack-2 のインスタンスのロールが STANDBY/PRIMARY モードであることを確認します。

```
show role instance-id 1
result "STANDBY"

show role instance-id 2
result "PRIMARY"
```

7. Rack-1 の instance-id 2 で、STANDBY へのロールのリセットを開始します。このステップにより、Rack-1 のロールが PRIMARY/STANDBY_ERROR から PRIMARY/STANDBY に移行します。

```
geo reset-role instance-id 2 role standby
```

8. Rack-1 のインスタンスのロールが PRIMARY/STANDBY であることを確認します。

```
show role instance-id 1
result "PRIMARY"

show role instance-id 2
result "STANDBY"
```

主要業績評価指標（KPI）

次の項では、KPI について説明します。

ETCD/Cachepod 複製 KPI

次の表に、ETCD/Cachepod 複製 KPI のリストを記載します。

表 *199 : geo_replication_total KPI*

KPI 名	説明	ラベル	有効な値
geo_replication _total	このKPIには、さまざまな同期タイプと複製タイプの複製要求/応答の総数が表示されます。	ReplicationRequest タイプ	要求/応答
		ReplicationSync タイプ	即時/保留/プル
		ReplicationNode	ETCD/CACHE_POD/PEER
		ReplicationReceiver	ローカル/リモート
		status	True/False
		status_code	エラーコード/説明

地理的なロール変更拒否の KPI

次の表に、地理的なロール変更拒否の KPI のリストを記載します。

表 *200 :* 地理的なロール変更拒否の *KPI*

KPI 名	説明	ラベル	有効な値
geo_RejectedRole Changed_total	このKPIには、STANDBYインスタンスで拒否された要求や受信したコールの合計数が表示されます。カウントの後、同じインスタンスがPRIMARYに移行されます。	RejectedCount	STANDBYインスタンスに対して受信された拒否されたコール/要求を示す数値。
		GRInstance 番号	1/2

モニタリング KPI

次の表に、モニタリング KPI のリストを記載します。

表 *201 : geo_monitoring_total KPI*

KPI 名	説明	ラベル	有効な値
geo_monitoring_total	このKPIには、ハートビート、remoteNotify、TriggerGR など、さまざまな種類の成功/失敗メッセージの総数が表示されます。	ControlAction タイプ	AdminMonitoring ActionType / AdminRemote MessageAction Type / AdminRole ChangeActionType
		ControlAction NameType	MonitorPod / MonitorBfd / RemoteMsgHeartbeat / RemoteMsgNotifyFailover / RemoteMsgNotify PrepareFailover / RemoteMsgGetSiteStatus / RemoteClusterPodFailure / RemoteSiteRole Monitoring / TriggerGRApi / ResetRoleApi
		管理者ノード	任意の文字列値。例：GRインスタンス識別子またはインスタンスキー/ポッド名
		ステータス コード	0/1001/1002/1003/ 1004/1005/1006/1007/ 1008/受信エラーコード（1206、1219、2404、...）
		ステータス メッセージ	

KPI 名	説明	ラベル	有効な値
			成功（0）/ STANDBY_ERROR => STANDBY/STANDBY => PRIMARY (0) / Pod Failure (0) / CLI (0) / BFD の障害 (0) / デコード失敗 (1001) / リモートステータスが使用不可 (1002) / ターゲットロールはサポートしていません (1002) / ポッド障害 (1002) / CLI (1002) / BFD の障害 (1002) / サイトがダウンしています (1003) / ポッド障害 (1003) / CLI (1003) / BFD の障害 (1003) / トラフィックヒット (1004) / ポッドエラー (1004) /CLI (1004) / BFD の障害 (1004) /現在のロールは以下ではありません STANDBY_ERROR/ STANDBY ロールを リセット (1005) / resetRole： etcd でキーが見つかりませんでした (1006) / ポッドあたりの モニタリングしきい値を超過しました (1007) / ハートビートの再試行の 失敗 (1008) / エラーメッセージを受信しました（この要求に使用可能なリモートホストがありません/選択されたリモートホスト<remotehostname>クライアント接続がありません/SLA がトランザクションに対して期限切れです/...）

表 *202 : geo_replication_finalpull_total KPI*

KPI 名	説明	ラベル名	有効な値
geo_replication _finalpulld_total	この KPI には、機能メッセージの最終取得時点で存在する Geo レプリケーションの総数が表示されます。	**MessageType**	これは要求または応答のメッセージタイプです。
		TotalTimeTaken	これは、リクエストの処理にかかった合計時間です。
		GRInstanceNumber	これは、次のリストからの GR インスタンス ID（数値）を表します。 • 1 • 2 • Instance.1 • Instance.2

BFD KPI

次の表に、BFD KPI のリストを記載します。

表 *203 : BFD KPI - 1*

KPI 名	説明	ラベル	有効な値
bgp_speaker _bfd_status	この KPI には、BGP スピーカーの BFD リンクステータスが表示されます。	status	STATE_UP / STATE_DOWN
geo_bfd status	この KPI には、Geo POD の BFD リンクステータスが表示されます。	status	STATE_UP / STATE_DOWN

表 *204 : BFD KPI - 2*

KPI 名	説明	ゲージ
bgp_speaker _bfd_status	この KPI には、BGP スピーカーの BFD リンクステータスが表示されます。	1（UP）または 0（DOWN）

KPI 名	説明	ゲージ
geo_bfd status	この KPI には、Geo POD の BFD リンクステータスが表示されます。	1（UP）または 0（DOWN）

クロスラックルーティング BFD インターフェイスのモニタリング

表 *205 :* クロスラックルーティング *BFD* インターフェイスのモニタリング *KPI*

KPI 名	説明	ラベル	有効な値
geo_monitoring total	この KPI は、ピアラックがダウンしている際に発生した Gateway Down または LocalBFDInterface Down メッセージの総数を、ゲートウェイ IP またはインターフェース名の詳細とともに表示します。	ControlAction タイプ	AdminMonitoring ActionType
		ControlAction NameType	MonitorGateway / MonitorLocalBfdInterface
		AdminNode	gateway_ip / interface_name
		status	すべての Proto ノードでゲートウェイ IP がダウンしています / すべての Proto ノードで BFD インターフェイスがダウンしています
		status_code	1012 / 1013
bgp_bfd_Monitor Interface_ status (Type - Gauge)	この KPI は、各ピアの接続ステータスを示します。この接続には BFD インターフェースが設定されており、リモートラック上のピアと接続されています。	interface	<ローカルラックインターフェイス名>
		peer_address	<リモートラックネイバーの IP アドレス>
		type	Bfd-Peer

KPI 名	説明	ラベル	有効な値
bgp_bfd_Monitor Remote_Rack_ status (Type - Gauge)	このKPIは、リモートラックのステータスを示します。現在のラックインターフェイスとリモートラックのピアは、BFDピアリングの一部として設定されます。両方のProtoノードからの接続のいずれかがUPの場合、ラックのステータスはUPになります。両方のProtoノードで接続がダウンしている場合、このKPIはリモートラックのステータスがDOWNであることを示します。	status	BFD_Remote Rack_STATUS

ローカル インターフェイス モニタリング

表 *206 :* ローカルインターフェイスのモニタリングの *KPI*

KPI 名	説明	ラベル	有効な値
geo_monitoring total	このKPIには、ローカルインターフェイスがダウンしたケースの総数と、インターフェイス名の詳細が表示されます。	ControlAction タイプ	AdminMonitoring ActionType
		ControlAction NameType	MonitorInterface
		AdminNode	interface_name
		status	すべてのProtoノードでローカルインターフェイスがダウンしています
		status_code	1014

GR インスタンス情報

表 *207 : GR* インスタンス情報の *KPI*

KPI 名	説明	ラベル	有効な値
gr_instance_ information (Type – Guage)	この KPI には、アプリケーションでの GR インスタンスの現在のロールが表示されます。	gr_instance_id	設定された GR インスタンス値（数値）

Geo メンテナンスモード

表 *208 : Geo* メンテナンスモードの *KPI*

KPI 名	説明	ラベル	有効な値
geo_MaintenanceMode info (Type – Guage)	この KPI には、ラックのメンテナンスモードの現在の状態が表示されます。	MaintenanceMode	0 = False 1 = True

バルク統計

次のセクションでは、GR 固有のバルク統計情報について詳しく説明します。

```
bulk-stats query GR-BGP-Incoming-Failed-Routes
 expression "sum(bgp_incoming_failedrouterequest_total) by (namespace, interface,
service_IP, next_hop, instance_id)"
 labels     [ instance_id interface next_hop service_IP ]
 alias      gr-bgp-routes-in
exit
bulk-stats query GR-Geo-Monitoring-Failure
 expression
"sum(geo_monitoring_total{ControlActionNameType=~'MonitorPod|RemoteMsgHeartbeat|
RemoteMsgGetSiteStatus|RemoteSiteRoleMonitoring|RemoteClusterPodFailure|RemoteMsgNotifyFailover|
RemoteMsgNotifyPrepareFailover|MonitorVip',status!~'success|monitoring.*'}) by (namespace,

AdminNode, ControlActionType, ControlActionNameType,  pod, status, status_code)"
 labels     [ pod AdminNode ControlActionNameType status status_code ]
 alias      gr-geo-monitoring-failure
exit
bulk-stats query GR-Geo-Monitoring-Success
 expression
"sum(geo_monitoring_total{ControlActionNameType=~'MonitorPod|RemoteMsgHeartbeat|
RemoteMsgGetSiteStatus|RemoteSiteRoleMonitoring|RemoteClusterPodFailure|RemoteMsgNotifyFailover|
RemoteMsgNotifyPrepareFailover',status=~'success|monitoring.*'}) by (namespace, AdminNode,

ControlActionType, ControlActionNameType,  pod, status)"
 labels     [ pod AdminNode ControlActionNameType status ]
 alias      gr-geo-monitoring
exit
bulk-stats query GR-Geo-Monitoring-Total
 expression
```

```
"sum(geo_monitoring_total{ControlActionNameType=~'MonitorPod|RemoteMsgHeartbeat|
RemoteMsgGetSiteStatus|RemoteSiteRoleMonitoring|RemoteClusterPodFailure|RemoteMsgNotifyFailover
|RemoteMsgNotifyPrepareFailover|MonitorVip'})
by (namespace, AdminNode, ControlActionType, ControlActionNameType,  pod, status)"
 labels     [ pod AdminNode ControlActionNameType status ]
 alias      gr-geo-monitoring
exit
bulk-stats query GR-Geo-Replication-Failure
 expression
"sum(geo_replication_total{ReplicationNode=~'CACHE_POD|ETCD|PEER',status!='success',
ReplicationRequestType='Response'}) by (namespace, ReplicationNode, ReplicationSyncType,
ReplicationReceiver,ReplicationRequestType,status,status_code)"
 labels     [ pod ReplicationNode ReplicationReceiver ReplicationRequestType
ReplicationSyncType status status_code ]
 alias      gr-geo-replication-failure
exit
bulk-stats query GR-Geo-Replication-Success
 expression "sum(geo_replication_total{ReplicationNode=~'CACHE_POD|ETCD|PEER',
status='success',ReplicationRequestType='Response'}) by (namespace, ReplicationNode,
ReplicationSyncType,ReplicationReceiver,ReplicationRequestType,status)"
 labels     [ pod ReplicationNode ReplicationReceiver ReplicationRequestType
ReplicationSyncType status ]
 alias      gr-geo-replication-success
exit
bulk-stats query GR-Geo-Replication-Total
 expression "sum(geo_replication_total{ReplicationNode=~'CACHE_POD|ETCD|PEER'})
by (namespace, ReplicationNode, ReplicationSyncType,ReplicationReceiver,
ReplicationRequestType,pod)"
 labels     [ pod ReplicationNode ReplicationReceiver ReplicationRequestType
ReplicationSyncType ]
 alias      gr-geo-replication-total
exit
bulk-stats query GR-Trigger-ResetRole-Api
 expression "sum(geo_monitoring_total{ControlActionNameType=~'TriggerGRApi|ResetRoleApi'})

by (namespace, AdminNode, ControlActionType, ControlActionNameType,  pod, status,
status_code)"
 labels     [ pod AdminNode ControlActionNameType status status_code ]
 alias      gr-api
exit
bulk-stats query GR-CDL-Index-Replication
 expression "sum(consumer_kafka_records_total) by (pod, origin_instance_id)"
 labels     [ origin_instance_id pod ]
 alias      gr-cdl-index-replication
exit
bulk-stats query GR-CDL-Inter-Rack-Replications-Failures
 expression "sum(datastore_requests_total{local_request='0',errorCode!='0'}) by
(operation,sliceName,errorCode)"
 labels     [ sliceName operation errorCode ]
 alias      gr-cdl-inter-rack-replications
exit
bulk-stats query GR-CDL-Inter-Rack-Replications-Success
 expression "sum(datastore_requests_total{local_request='0',errorCode='0'}) by
(operation,sliceName,errorCode)"
 labels     [ sliceName operation errorCode ]
 alias      gr-cdl-inter-rack-replications
exit
bulk-stats query GR-CDL-Inter-Rack-Replications-Total
 expression "sum(datastore_requests_total{local_request='0'}) by
(operation,sliceName,errorCode)"
 labels     [ sliceName operation errorCode ]
 alias      gr-cdl-inter-rack-replications
exit
bulk-stats query GR-CDL-Intra-Rack-Operations-Failures
```

```
 expression "sum(datastore_requests_total{local_request='1',errorCode!='0'}) by
(operation,sliceName,errorCode)"
 labels     [ sliceName operation errorCode ]
 alias      gr-cdl-intra-rack-operations
exit
bulk-stats query GR-CDL-Intra-Rack-Operations-Success
 expression "sum(datastore_requests_total{local_request='1',errorCode='0'}) by
(operation,sliceName,errorCode)"
 labels     [ sliceName operation errorCode ]
 alias      gr-cdl-intra-rack-operations
exit
bulk-stats query GR-CDL-Intra-Rack-Operations-Total
 expression "sum(datastore_requests_total{local_request='1'}) by
(operation,sliceName,errorCode)"
 labels     [ errorCode operation sliceName ]
 alias      gr-cdl-intra-rack-operations
exit
bulk-stats query GR-CDL-Session-Count-Per-Slice
 expression
sum(avg(db_records_total{namespace=~'$namespace',session_type='total'})by(systemId,sliceName))by(sliceName)

 labels     [ sliceName ]
 alias      gr-cdl-session-count-per-slice
exit
bulk-stats query GR-CDL-Session-Count-Per-System-ID
 expression sum(avg(db_records_total{namespace=~'$namespace',session_type='total'})
by(systemId,sliceName))by(systemId)
 labels     [ systemId ]
 alias      gr-cdl-session-count-per-system-id
exit
bulk-stats query GR-CDL-Slot-Records-Per-Slice
 expression "sum(slot_records_total{pod=~'.*',systemId!=''}) by (pod, sliceName)"
 labels     [ pod sliceName ]
 alias      gr-cdl-slot-records-per-slice
exit
bulk-stats query GR-CDL-Slot-Records-Per-System-ID
 expression "sum(slot_records_total{pod=~'.*',systemId!=''}) by (pod, systemId)"
 labels     [ pod systemId ]
 alias      gr-cdl-slot-records-per-system-id
exit
bulk-stats query GR-CDL-Total-Session-Count
 expression "sum(db_records_total{namespace=~'$namespace',session_type='total'}) by
(systemId,sliceName)"
 labels     [ sliceName systemId ]
 alias      gr-cdl-total-session-count
exit
```

GR 関連の統計情報の詳細については、次を参照してください。

- cnSGW-C 統計情報では、gr_instance_id ラベルを使用して GR 固有の統計情報をフィルタリングできます。

  詳細については、「*UCC* サービング ゲートウェイ コントロール プレーンの機能 - メトリックリファレンス」を参照してください。

アラート

ここでは、GR アラートについて詳しく説明します。

BFD アラート

次の表に、interval-seconds が 60 に設定されているルールグループ BFD のアラートを示します。

表 *209 :* アラートルールグループ：***BFD***

アラートルール	重大度	期間（分単位）	タイプ
BFD-Link-Fail	critical	1	通信アラーム
	式：sum by (namespace,pod,status) (bgp_speaker_bfd_status{status='BFD_STATUS'}) == 0 **説明**：このアラートは、BGP ピアリングに関連付けられた BFD リンクがダウンしている場合に生成されます。		

GR アラート

次の表に、interval-seconds が 60 に設定されているルールグループ GR のアラートを示します。

表 *210 :* アラートルールグループ：***GR***

アラートルール	重大度	期間（分単位）	タイプ
Cache-POD-Replication-Immediate-Local	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='CACHE_POD', ReplicationSyncType='Immediate',ReplicationReceiver='local', ReplicationRequestType='Response',status='success'}[1m]))/sum by (namespace) (increase(geo_replication_total{ReplicationNode='CACHE_POD', ReplicationSyncType='Immediate',ReplicationReceiver='local', ReplicationRequestType='Request'}[1m])))*100 < 90 **説明**：このアラートは、CACHE_POD sync type:Immediate および replication receiver:Local の成功率がしきい値未満の場合に生成されます。		

アラートルール	重大度	期間（分単位）	タイプ
Cache-POD-Replication-Immediate-Remote	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='CACHE_POD', ReplicationSyncType='Immediate',ReplicationReceiver='remote', ReplicationRequestType='Response',status='success'}[1m]))/sum by (namespace) (increase(geo_replication_total{ReplicationNode='CACHE_POD', ReplicationSyncType='Immediate',ReplicationReceiver='remote', ReplicationRequestType='Request'}[1m])))*100 < 90 **説明**：このアラートは、CACHE_POD sync type:Immediate および replication receiver:Remote の成功率がしきい値未満の場合に生成されます。		
Cache-POD-Replication-PULL-Remote	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='CACHE_POD', ReplicationSyncType='PULL',ReplicationReceiver='remote', ReplicationRequestType='Response',status='success'}[1m]))/sum by (namespace) (increase(geo_replication_total{ReplicationNode='CACHE_POD', ReplicationSyncType='PULL',ReplicationReceiver='remote', ReplicationRequestType='Request'}[1m])))*100 < 90 **説明**：このアラートは、CACHE_POD sync type:PULL および replication receiver:Remote の成功率がしきい値未満の場合に生成されます。		

アラートルール	重大度	期間（分単位）	タイプ
ETCD-Replication-Immediate-Local	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='ETCD', ReplicationSyncType='Immediate',ReplicationReceiver='local', ReplicationRequestType='Response',status='success'}[1m]))/sum by (namespace) (increase(geo_replication_total{ReplicationNode='ETCD', ReplicationSyncType='Immediate',ReplicationReceiver='local', ReplicationRequestType='Request'}[1m])))*100 < 90 **説明**：このアラートは、ETCD sync type:Immediate および replication receiver:Local の成功率がしきい値未満の場合に生成されます。		
ETCD-Replication-Immediate-Remote	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='ETCD', ReplicationSyncType='Immediate',ReplicationReceiver='remote', ReplicationRequestType='Response',status='success'}[1m]))/sum by (namespace) (increase(geo_replication_total{ReplicationNode='ETCD', ReplicationSyncType='Immediate',ReplicationReceiver='remote', ReplicationRequestType='Request'}[1m])))*100 < 90 **説明**：このアラートは、ETCD sync type:Immediate および replication receiver:Remote の成功率がしきい値未満の場合に生成されます。		
ETCD-Replication-PULL-Remote	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='ETCD', ReplicationSyncType='PULL',ReplicationReceiver='remote', ReplicationRequestType='Response',status='success'}[1m]))/ sum by (namespace) (increase(geo_replication_total {ReplicationNode='ETCD',ReplicationSyncType='PULL', ReplicationReceiver='remote',ReplicationRequestType= 'Request'}[1m])))*100 < 90 **説明**：このアラートは、ETCD sync type:PULL および replication receiver:Remote の成功率がしきい値未満の場合に生成されます。		

アラートルール	重大度	期間（分単位）	タイプ
Heartbeat-Remote -Site	critical	-	通信アラーム
	式：sum by (namespace) (increase(geo_monitoring_total{ControlActionNameType= 'RemoteMsgHeartbeat',status!='success'}[1m])) > 0 **説明**：このアラートは、リモートサイトへの定期的なハートビートが失敗するとトリガーされます。		
Local-Site-POD-Monitoring	critical	-	通信アラーム
	式：sum by (namespace,AdminNode) (increase(geo_monitoring_total{ControlActionNameType ='MonitorPod'}[1m])) > 0 **説明**：このアラートは、ローカルサイトのポッドモニタリングに障害が発生した場合、Label:{{$labels.AdminNode}} に記載されているポッドの設定されたしきい値を超えた場合にトリガーされます。		
PEER-Replication-Immediate-ローカル	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='PEER', ReplicationSyncType='Immediate',ReplicationReceiver='local', ReplicationRequestType='Response',status='success'} [1m]))/sum by (namespace) (increase(geo_replication_total {ReplicationNode='PEER',ReplicationSyncType= 'Immediate',ReplicationReceiver='local', ReplicationRequestType='Request'}[1m])))*100 < 90 **説明**：このアラートは、PEER sync type:Immediate および replication receiver:Local の成功率がしきい値未満の場合に生成されます。		

アラートルール	重大度	期間（分単位）	タイプ
PEER-Replication-Immediate-リモート	critical	1	通信アラーム
	式：(sum by (namespace) (increase(geo_replication_total{ReplicationNode='PEER', ReplicationSyncType='Immediate',ReplicationReceiver='remote', ReplicationRequestType='Response',status='success'} [1m]))/sum by (namespace) (increase(geo_replication_total {ReplicationNode='PEER',ReplicationSyncType='Immediate', ReplicationReceiver='remote',ReplicationRequestType= 'Request'}[1m])))*100 < 90 **説明**：このアラートは、PEER sync type:Immediate および replication receiver:Remote の成功率がしきい値未満の場合に生成されます。		
RemoteCluster-PODFailure	critical	-	通信アラーム
	式：sum by (namespace,AdminNode) (increase(geo_monitoring_total{ControlActionNameType ='RemoteClusterPodFailure'}[1m])) > 0 **説明**：このアラートは、Label:{{$labels.AdminNode}} に記載されているポッドのリモートサイトでポッド障害が検出された場合に生成されます。		
RemoteMsg NotifyFailover	critical	1	通信アラーム
	式：sum by (namespace,status) (increase(geo_monitoring_total{ControlActionNameType ='RemoteMsgNotifyFailover',status!='success'}[1m])) > 0 **説明**：このアラートは、Label:{{$labels.status}} に記載されている理由で一時的なロール RemoteMsgNotifyFailover が失敗した場合に生成されます。		
RemoteMsg NotifyPrepare フェールオーバー	critical	1	通信アラーム
	式：sum by (namespace,status) (increase(geo_monitoring_total{ControlActionNameType ='RemoteMsgNotifyPrepareFailover',status!='success'}[1m])) > 0 **説明**：このアラートは、Label:{{$labels.status}} に記載されている理由で一時的なロール RemoteMsgNotifyPrepareFailover が失敗した場合に生成されます。		

アラートルール	重大度	期間（分単位）	タイプ
RemoteSite-RoleMonitoring	critical	-	通信アラーム
	式：sum by (namespace,AdminNode) (increase(geo_monitoring_total{ControlActionNameType ='RemoteSiteRoleMonitoring'}[1m])) > 0 **説明**：このアラートは、RemoteSiteRoleMonitoring がパートナーラックのインスタンスのロールの不整合を検出し、その結果、ローカルラックの各インスタンスのロールをプライマリに変更した場合に生成されます。影響を受けるインスタンスは Label:{{$labels.AdminNode}} にあります。		
ResetRoleApi-Initiated	critical	-	通信アラーム
	式：sum by (namespace,status) (increase(geo_monitoring_total{ControlActionNameType ='ResetRoleApi'}[1m])) > 0 **説明**：このアラートは、Label:{{$labels.status}} に記載されているロールの状態遷移とともに ResetRoleApi が開始されたときに生成されます。		
TriggerGRApi-Initiated	critical	-	通信アラーム
	式：sum by (namespace,status) (increase(geo_monitoring_total{ControlActionNameType ='TriggerGRApi'}[1m])) > 0 **説明**：このアラートは、Label:{{$labels.status}} に記載されている理由で TriggerGRApi が開始されたときに生成されます。		
VIP-Monitoring-Failures	critical	-	通信アラーム
	式：sum by (namespace,AdminNode) (increase(geo_monitoring_total{ControlActionNameType ='MonitorVip'}[1m])) > 0 **説明**：このアラートは、Label:{{$labels.AdminNode}} に記載されている VIP とインスタンスの VIP モニタリング障害の検出時に GR が生成されると生成されます。		

CDL アラート

次の表に、interval-seconds が 60 に設定されているルールグループ CDL のアラートを示します。

表 *211 :* アラートルールグループ：***CDL***

アラートルール	重大度	期間（分単位）	タイプ
GRPC-Connections-Remote-Site	critical	1	通信アラーム
	式：sum by (namespace, pod, systemId) (remote_site_connection_status) !=4 **説明**：このアラートは、リモートサイトへの GRPC 接続数が 4 ではない場合に生成されます。		
Inter-Rack -CDL-Replication	critical	1	通信アラーム
	式：(sum by (namespace) (increase(datastore_requests_total{local_request=\"0\", errorCode=\"0\"}[1m]))/sum by (namespace) (increase(datastore_requests_total{local_request=\"0\"} [1m])))*100 < 90 **説明**：このアラートは、ラック間CDL レプリケーションの成功率がしきい値未満の場合に生成されます。		
Intra-Rack -CDL-Replication	critical	1	通信アラーム
	式：(sum by (namespace) (increase(datastore_requests_total{local_request=\"1\", errorCode=\"0\"}[1m]))/sum by (namespace) (increase(datastore_requests_total{local_request=\"1\"} [1m])))*100 < 90 **説明**：このアラートは、ラック内CDL レプリケーションの成功率がしきい値未満の場合に生成されます。		

メンテナンス モード

cnSGW-Cはメンテナンスモードフラグをサポートしており、GRセットアップのクラスタがインサービス（ローリングアップグレード）用にスケジュールされている場合に、クラスタへの影響を無効化できます。これは、他の従属クラスタがその役割やその他のアクティビティを、ターゲットのクラスタで問題なく実行できるようにするのに役立ちます。

メンテナンスモードフラグが **true** に設定されている場合、クラスタロールの変更およびラックに対する GR トリガーは、CLI ベースのフェールオーバーの場合にのみ許可されます。

実行中、すべてのモニタリングスレッドでフラグのランタイム値がチェックされ、メンテナンスモードフラグが **true** に設定されている場合は実行が保留されます。デフォルトでは、新規

インストールの場合、フラグは **false** に設定されます。要件に基づいてメンテナンスモードを設定するには、次の設定を使用します。

```
config
  geo maintenance mode { true | false }
  end
```

注：

- **geo maintenance mode { true | false }**：メンテナンスモードを有効/無効にします。

  メンテナンスモードの値は、etcd に保存されます。

両方のクラスタを同時にメンテナンスできます。従属クラスタがすでにメンテナンス中の場合は、システムをメンテナンスモードに移行できます。メンテナンス作業を開始する前に、`geo maintenance mode` フラグの値を **true** に設定してください。メンテナンスが完了したら、システムの正常性を確認後、フラグを **false** にリセットします。

メンテナンスフラグが true に設定されている場合：

- すべてのモニタリングアクティビティが一時停止します。
- 従属クラスタは、ローカルの障害を検出してもフェールオーバーをトリガーできません。
- レプリケーション アクティビティはクラスタで続行されます。
- メンテナンスモードでは、サイトのインスタンスロールは暗黙的に変更されませんが、ロールの変更は `geo switch-role role` CLI コマンドを使用して可能です。

メンテナンス中のクラスタとの間で GR トリガーの送受信はできません。メンテナンス中のクラスタからは、CLI ベースのフェールオーバーのみがサポートされます。メンテナンスモードを無効にしたら、ポッドと VIP のモニタリング用の新しいデータから開始します。リモートクラスタには、`NotifyMaintenanceActivity() [Operation 24]` メッセージを使用してメンテナンスモード値が通知されます。

例

次に、設定例を示します。

```
geo maintenance mode true
result "success"

geo maintenance mode false
result "success"
```

メンテナンスモードのステータスの表示

メンテナンスモードのステータスを確認するには、次の `show` コマンドを使用します。

```
show geo maintenance mode
result "geo maintenance mode is disabled"
```

第 43 章

サービス構成の拡張機能

- 機能の概要と変更履歴 （659 ページ）
- 機能説明 （660 ページ）
- 機能設定 （660 ページ）
- トラブルシューティング情報 （666 ページ）

機能の概要と変更履歴

要約データ

表 *212* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *213* : マニュアルの変更履歴

改訂の詳細	リリース
DNN リストのサポートを追加	2021.02.3
初版	2020.01.0

機能説明

cnSGW-C は、サブスクライバマップとオペレータポリシーの設定をサポートします。

SGW プロファイルは、SGW サービスまたはノードを表します。オペレータポリシーは、サブスクライバポリシーの関連付けに基づいて決まります。

機能設定

この機能の設定には、次の手順が含まれます。

- SGW プロファイル。詳細については、SGW プロファイルの設定 （660 ページ）を参照してください。
- サブスクライバポリシー。詳細については、サブスクライバポリシーの構成（661 ページ）を参照してください。
- オペレータポリシー。詳細については、オペレータポリシーの設定 （662 ページ）を参照してください。
- ポリシー DNN。詳細については、ポリシー DNN の設定 （663 ページ）を参照してください。

SGW プロファイルの設定

この機能を設定するには、次の設定を使用します。

```
config
   profile sgw sgw_name
      locality locality_code
      fqdn dnn_name
      subscriber-policy policy_name
      end
```

注：

- **locality** *locality_code*：地域コードを指定します。文字列で指定する必要があります。
- **fqdn** *dnn_name*：cnSGW-C FQDN を指定します。
- **subscriber-policy** *policy_name*：サブスクライバポリシー名を指定します。文字列で指定する必要があります。

設定例

次に、設定例を示します。

```
config
   profile sgw sgw-data
```

```
locality LOC1
fqdn 209.165.200.254
subscriber-policy subpol1
end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw
profile sgw sgw-data
locality LOC1
fqdn 209.165.200.254
subscriber-policy subpol1
```

サブスクライバポリシーの構成

（注） サポートされている最大サブスクライバ マッピング プロファイル数は 64 です。

この機能を設定するには、次の構成を使用します。

```
config
   policy subscriber subscriber_name
      precedence precedence_value
         imsi
            mcc mcc_value
            mnc mnc_value
            msin first_value last_value
         serving-plmn
            mcc mcc_value
            mnc mnc_value
         imsi-start-range range_value
         imsi-stop-range range_value
         supi-start-range range_value
         supi-stop-range range_value
         operator-policy policy_name
         end
```

（注） パラメータはすべてオプションです。

注：

- **precedence** *precedence_value*：エントリの優先順位を指定します。1 ～ 2048 の範囲の整数である必要があります。
- **mcc** *mcc_value*：モバイル国コード（MCC）を指定します。3 桁の整数である必要があります。

- **mnc** *mnc_value*：モバイルネットワークコード（MNC）を指定します。2 桁または 3 桁の整数である必要があります。
- **msin** *first_value last_value*：モバイルサブスクライバ識別番号（MSIN）の範囲を指定します。

  first_value：MSIN 範囲の開始値を指定します。1 ～ 9999999999 の範囲の整数である必要があります。

  last_value：MSIN 範囲の終了値を指定します。1 ～ 9999999999 の範囲の整数である必要があります。
- **operator-policy** *policy_name*：オペレータポリシー名を指定します。文字列である必要があります。
- **imsi-start-range** *range_value*：IMSI の開始範囲を指定します。1000000000000000 ～ 9999999999999999 の範囲の整数である必要があります。
- **imsi-stop-range** *range_value*：IMSI 停止範囲を指定します。1000000000000000 ～ 9999999999999999 の範囲の整数である必要があります。
- **supi-start-range** *range_value*：SUP 開始範囲を指定します。1000000000000000 ～ 999999999999999 の範囲の整数である必要があります。
- **supi-stop-range** *range_value*：SUPI 停止範囲を指定します。1000000000000000 ～ 999999999999999 の範囲の整数である必要があります。

設定例

次に、設定例を示します。

```
config
   policy subscriber sub1
      precedence 2
         imsi mcc 123 mnc 456
         imsi msin first 99 last 100
      serving-plmn mcc 404 mnc 678
      supi-start-range 100000000000001
      supi-stop-range  199999999999999
      imsi-start-range 200000000000001
      imsi-stop-range  299999999999999
      operator-policy op1
      end
```

オペレータポリシーの設定

（注） サポートされるオペレータ ポリシー プロファイルの最大数は 1000 です。

この機能を設定するには、次の設定を使用します。

```
config
  policy operator operator_name
  policy dnn dnn_policy_name
  end
```

注：

- **policy operator** *operator_name*：オペレータポリシー名を指定します。文字列で指定する必要があります。
- **policy dnn** *dnn_policy_name*：DNN ポリシー名を指定します。文字列で指定する必要があります。

設定例

次に、設定例を示します。

```
config
   policy operator op1
   policy dnn pol1
   end
```

ポリシー DNN の設定

ここでは、ポリシー DNN を設定し、それを cnSGW-C に追加する方法について説明します。DNN サポートにより、APN 名、APN ネットワーク識別子、および APN オペレータ識別子に従って、正確な APN プロファイルを判断できます。

（注） サポートされる DNN ポリシーの最大数は 1000 です。

この機能を設定するには、次の構成を使用します。

```
config
   policy dnn dnn_policy_name
       dnn dnn_name
          dnn-list dnn_list
          profile profile_name
       dnn network-identifier network_identifier_name operator-identifier
operator_identifier_name profile profile_name
       dnn operator-identifier operator_identifier_name profile profile_name
       dnn operator-identifier profile profile_name
       end
```

注：

- **dnn** *dnn_name*：DNN 名を指定します。
- **network-identifier** *network_identifier_name*：ネットワーク識別子を指定します。文字列である必要があります。
- **profile** *profile_name*：プロファイル名を指定します。文字列である必要があります。

- **operator-identifier** *operator_identifier_name*：オペレータ識別子を指定します。文字列である必要があります。
- **profile** *default_dnn_profile*：デフォルトの DNN プロファイル名を指定します。
- **dnn-list** *dnn_list*：DNN プロファイルを選択するための DNN リストを指定します。これにより、複数の DNN を単一のプロファイルに関連付けるために必要なエントリ数を最小限に抑えることができます。

（注） 現在の評価基準に基づいて、関連付けられたプロファイルを選択する際の一致順序は次のとおりです。

- DNN/DNN リスト
- NI + OI
- NI
- OI
- デフォルト

重複する基準は設定しないでください。

設定例

次に、設定例を示します。

```
config
   policy dnn polsub1
      dnn network-identifier ims profile ims1
      dnn network-identifier ims operator-identifier ims.com profile ims
      dnn network-identifier voice operator-identifier volte profile voiceprofile
      dnn operator-identifier data profile data-profile
   profile default-dnn-profile
   end

config
   policy dnn polsub1
      dnn intershat dnn-list [ intershat1 intershat2 intershat3 ]
         profile profile_name
            dnn network-identifier ims operator-identifier ims.com profile ims
            dnn operator-identifier volte profile voiceprofile
            dnn operator-identifier data profile profile_name
            end
```

設定の変更による影響

このセクションでは、既存のコール、新しいPDN、または新しいサブスクライバに対する構成の変更の影響または動作について説明します。

変更内容	cnSGW-C の既存コール	cnSGW-C の新しい PDN または新しいサブスクライバ
新しい SGW プロファイルの定義と古いプロファイルの削除（ポッドの再起動の有無は問わない）	新しいトランザクションが発生すると、コールは CDL から cn-SGW-C にロードされます。 観測される結果： cnSGW-C プロファイル構成の変更とコールの終了。	新しい PDN を拒否し、既存のコールを削除します。 新しいサブスクライバは新しい SGW プロファイルを使用します。
サブスクライバマップの削除	影響なし	次に対して変更された構成を適用します。 • サブスクライバポリシー • オペレータポリシー • DNN ポリシー • DNN プロファイル
SGW サービスでのサブスクライバマップの変更	影響なし	次に対して変更された構成を適用します。 • サブスクライバポリシー • オペレータポリシー • DNN ポリシー • DNN プロファイル
オペレータポリシーの削除	影響なし	次に対して変更された構成を適用します。 • サブスクライバポリシー • オペレータポリシー • DNN ポリシー • DNN プロファイル
削除されたオペレータポリシーまたは新しいオペレータポリシーの追加	影響なし	次に対して変更された構成を適用します。 • サブスクライバポリシー • オペレータポリシー • DNN ポリシー • DNN プロファイル

変更内容	cnSGW-C の既存コール	cnSGW-C の新しい PDN または新しいサブスクライバ
サブスクライバマップのオペレータポリシー名の変更	影響なし	次に対して変更された構成を適用します。 • サブスクライバポリシー • オペレータポリシー • DNN ポリシー • DNN プロファイル

トラブルシューティング情報

ここでは、サービス設定の問題を解決するためのトラブルシューティング情報について説明します。

設定エラー

このセクションでは、サービス構成中に cnSGW-C が報告するエラーについて説明します。

サブスクライバポリシー構成エラー

```
show config-error
ERROR
COMPONENT ERROR DESCRIPTION
--------------------------------------------------------------------------
SGWProfile Subscriber policy name : polSubSgw in profile sgw1 is not configured
```

オペレータポリシー構成エラー

```
show config-error
ERROR COMPONENT ERROR DESCRIPTION
--------------------------------------------------------------------------
SubscriberPolicy Operator policy : op2 under subscriber policy polSubSgw is not configured
```

DNN ポリシー構成エラー

```
show config-error
ERROR COMPONENT ERROR DESCRIPTION
--------------------------------------------------------------------------
OperatorPolicy Dnn policy name : dnn_1 in operator policy op1 is not configured
```

第 44 章

SGW 変更サポート

- 機能の概要と変更履歴 （667 ページ）
- 機能説明 （668 ページ）
- 機能の仕組み （670 ページ）
- 機能設定 （686 ページ）
- cnSGW-CDR でサポートされている CDR フィールド （698 ページ）
- SGW 変更 OAM サポート （716 ページ）

機能の概要と変更履歴

要約データ

表 *214*：要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *215*：マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.02.0

機能説明

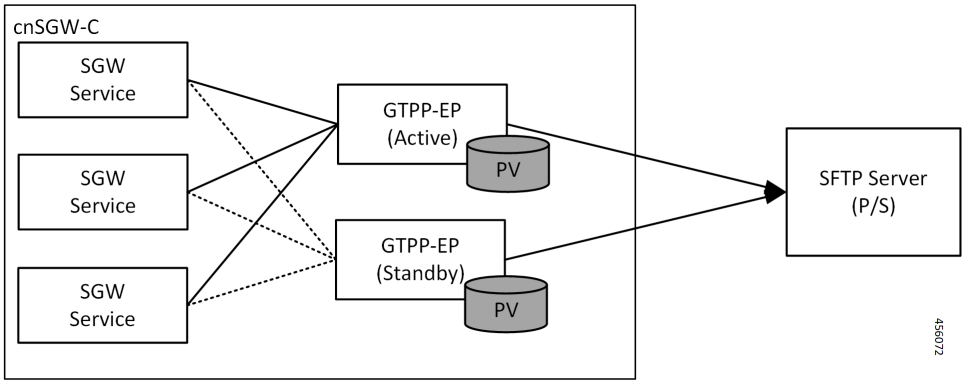
表 *216 :* 機能の履歴

機能名	リリース情報	説明
コンバージドコールの課金サポート	2024.02.0	UPF がコラップスドデータパス機能に対応しているため、cnSGW はコンバージド UPF（UPF + SGW-U）による課金をサポートします。このサポートにより、キャリアが SGW ベースの CDR を使用して報告と課金を行うローカルブレイクアウト（LBO）コールへの影響を防ぎます。 **デフォルト設定**：該当なし

cnSGW-c は次をサポートしています。

- GTPP 課金（Gz）インターフェイス
- 課金データレコード（CDR）のサブスクライバの監視
- CDR ディクショナリ：**custom24**
- 2 つのカスタムファイル形式：**custom1**（デフォルト）および **custom5**
- アクティブモードまたはスタンバイモードで機能する GTPP-EP ポッドの 1 つのレプリカ（GTPP-EP ポッドが 1 つのインスタンスで設定されている場合、2 つのポッドが生成されます）
- ローカルファイルシステムへの CDR レコードの書き込み
- 統合 UPF（UPF + SGW-U）による課金

アーキテクチャ



- GTP（GTP Prime）または GTPP-EP は新しいエンドポイントポッドであり、SFTP 経由で仲介サーバーまたは CGF サーバーとインターフェイスします。
- GTP はローカルディスク（永続ボリューム）に接続します。この接続は、各サーバーまたは仮想マシン（VM）に付属します。
- SGW サービスが CDR を生成し、ストレージのために GTP エンドポイントにレコードを送信します。
- GTP は永続ストレージのフラットファイルに ASN.1 エンコーディングで CDR を保存します。
- GTP は、仲介サーバーや CGF に SCTP を介してフラットファイルをプッシュします。

課金機能は 2 つの部分に分かれています。

- アカウンティングおよび CDR 生成：
  - SGW サービスは、SGW-U を使用して Sxa インターフェイスで確立された各ベアラーの使用状況レポートルール（URR）を生成します。
  - SGW サービスは、トリガーイベントでレポートされた使用状況情報を使用して、アカウンティング情報を生成します。
- CDR の管理とストレージ：
  - GTPP-EP マイクロサービスまたは K8 ポッドが CDR をアーカイブし、CDR ファイルを外部ストレージサーバーにプッシュします。
  - GTPP-EP は、ストリーミング GRPC IPC エンドポイントを介して SGW サービスから proto-CDR を受信します。
  - GTPP-EP は、受信した各 proto-CDR を（CLI を使用して）ディクショナリで指定されている ASN.1 形式にエンコードします。
  - ASN.1 CDR は、CLI 設定で指定されたパターンでフラットファイルに書き込まれます。

- 定期的に SFTP プロトコルを使用して新しい CDR ファイルを、設定されている外部ストレージサーバーに転送します。

ローミングのサポート

ローミングシナリオでは、Gz インターフェイスとオフラインアカウンティング機能を使用して、CDR レコードを外部 PGW と照合します。

オペレータポリシーは、SGW プロファイルに記述されているサブスクライバのサブセットの動作を変更するメカニズムを提供します。cnSGW-C は、オペレータポリシーの下で呼制御プロファイルをサポートし、アカウンティングモードを制御（課金の有効化または無効化）し、追加の課金設定を定義します。

デフォルトのアカウンティングモードは [NONE] で、課金が無効であることを示します。

呼制御プロファイルのアカウンティングモード値で、SGW プロファイルの設定値がオーバーライドされます。

次の設定の詳細を参照してください。

- 呼制御プロファイルの設定
- 呼制御プロファイルでの課金の特性

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

初期接続での URR のインストールコールフロー

ここでは、初期接続での URR のインストールコールフローについて説明します。

図 *112 :* 初期接続での *URR* のインストールコールフロー

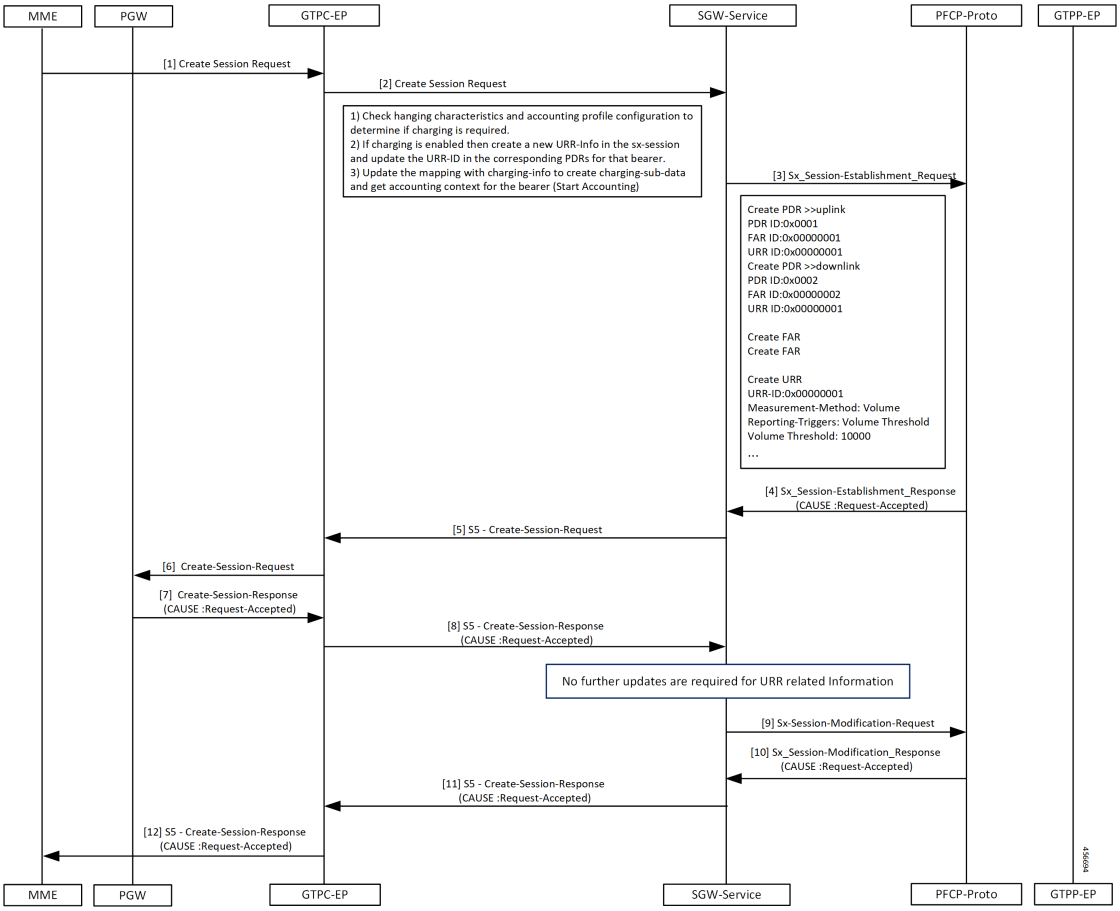


表 *217 :* 初期接続での *URR* のインストールコールフローの説明

ステップ	説明
1	MME が Create Session Request を GTPC-EP に送信します。
2	GTPC-EP が Create Session Request を SGW サービスポッドに転送します。
3	SGW サービスポッドが Sx Session Establishment Request を PFCP プロトコルに送信します。
4	PFCP プロトコルが Request-Accepted の理由で Sx Session Establishment Response を SGW サービスに送信します。
5	SGW サービスポッドが S5 Create Session Request を GTPC-EP に送信します。
6	GTPC-EP が S5 Create Session Request を PGW に送信します。
7	PGW が Request-Accepted の理由で Create Session Response を GTPC-EP に送信します。

ステップ	説明
8	GTPC-EP が Request-Accepted の理由で S5 Create Session Response を SGW サービスに送信します。
9	SGW サービスポッドが Sx Session Modification Request を PFCP プロトコルに送信します。
10	SGW サービスポッドが Request-Accepted の理由で PFCP プロトコルから Sx Session Modification Response を受信します。
11	SGW サービスポッドが Request-Accepted の理由で S5 Create Session Response を GTPC-EP に送信します。
12	GTPC-EP が Request-Accepted の理由で S5 Create Session Response を MME に転送します。

SGW CDR コールフロー

ここでは、SGW CDR コールフローについて説明します。

図 *113 : SGW CDR* コールフロー

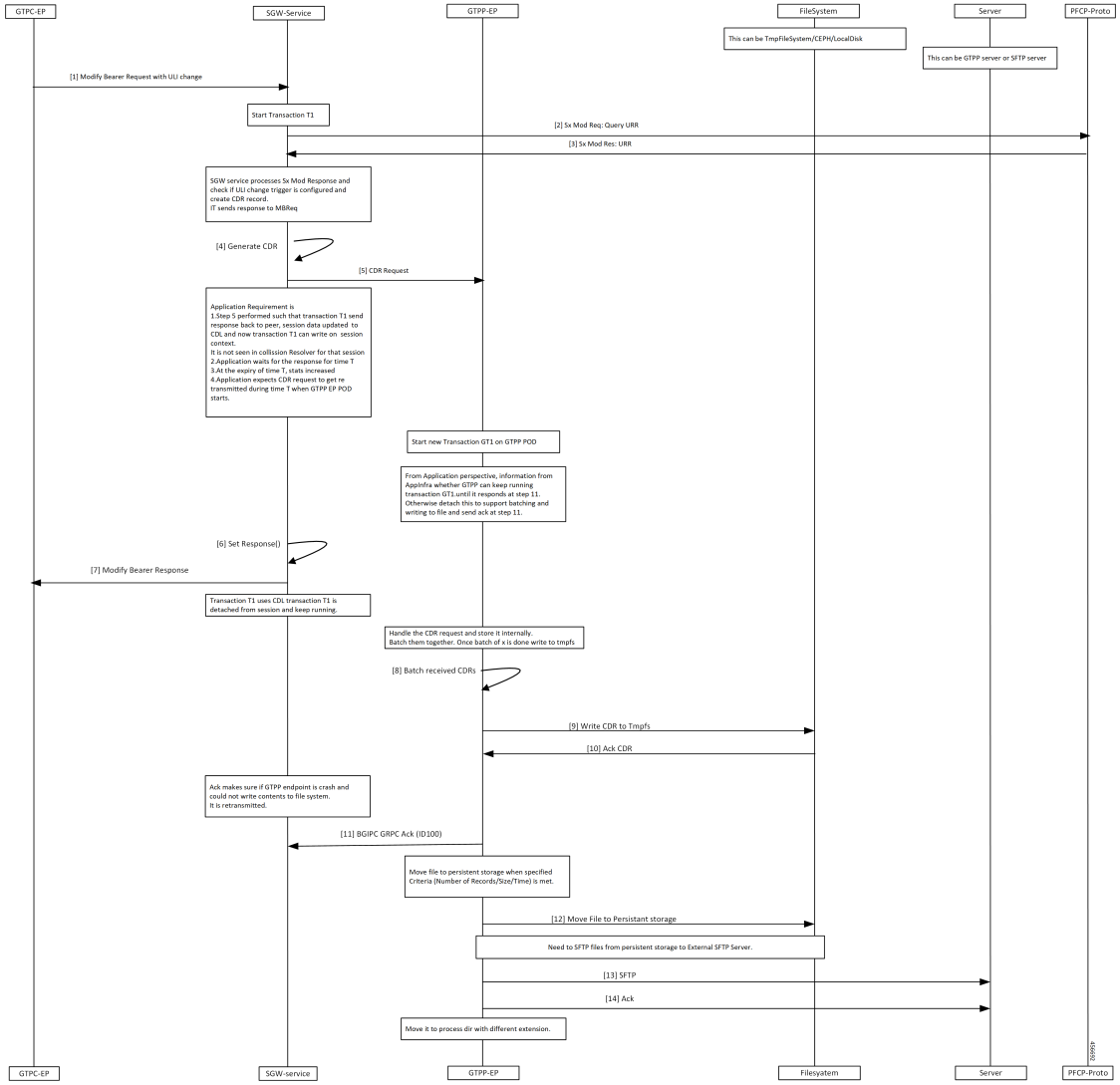


表 *218 : SGW CDR* コールフローの説明

ステップ	説明
1	GTPC-EP が URI を使用して Modify Bearer Request を SGW サービスポッドに送信します。
2	SGW サービスポッドがクエリ URR を含む Sx Mod Request を PFCP プロトコルに送信します。
3	SGW サービスポッドが URR を含む Sx Mod Response を PFCP プロトコルから受信します。

ステップ	説明
4	SGW サービスポッドが CDR を生成します。
5	SGW サービスポッドが CDR 要求を GTPP-EP に送信します。
6	SGW サービスポッドが set response () 関数をトリガーします。
7	SGW サービスポッドが Modify Bearer Response を GTPC-EP に送信します。
8	GTPP-EP が受信した CDR 要求をバッチ処理します。
9	GTPP-EP がバッチ処理された CDR 要求を TmpF5 ファイルに送信します。
10	GTPP-EP がファイルシステムから CDR ACK を受信します。
11	GTPP-EP が GRPC ACK を SGW サービスポッドに送信します。
12	指定された基準（レコード数、サイズ、または時間）を満たすと、GTPP-EP がファイルを永続ストレージに移動します。
13	GTPP-EP が永続ストレージから SFTP ファイルをサーバーに送信します。
14	GTPP-EP がサーバーから ACK を受信し、異なる拡張子を付けてプロセスディレクトリに移動します。

切断時の URR の削除と CDR レポートのコールフロー

ここでは、切断時の URR の削除と CDR レポートのコールフローについて説明します。

図 *114* : 切断時の *URR* の削除と *CDR* レポートのコールフロー

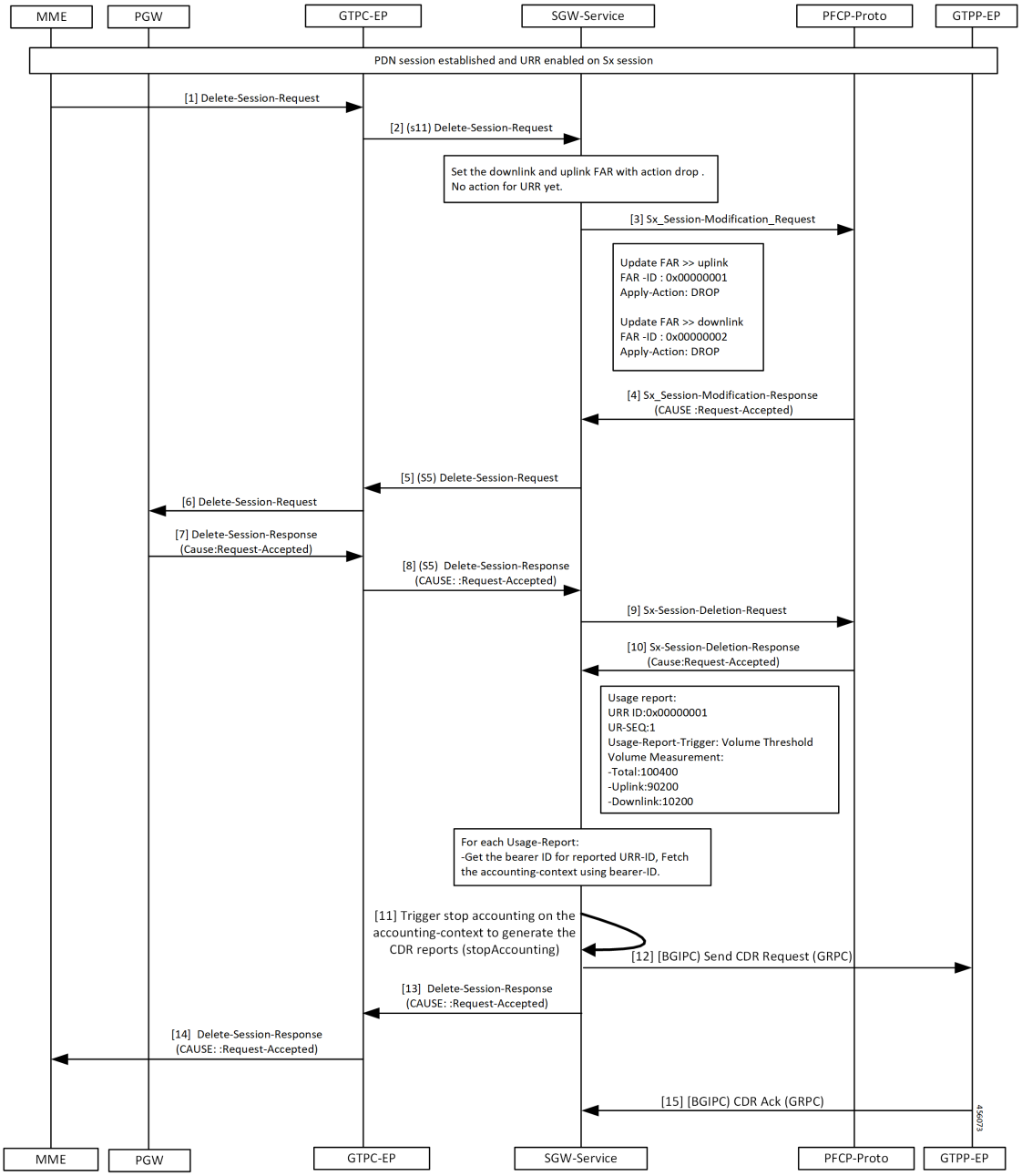


表 *219* : 切断時の *URR* の削除と *CDR* レポートのコールフローの説明

ステップ	説明
1	PDN セッションが確立され、URR が Sx セッションに対して有効になります。 MME が GTPC-EP にセッション削除要求を送信します。

ステップ	説明
2	GTPC-EP が SGW サービスポッドに S11 ベアラー削除要求を転送します。
3	SGW サービスポッドが Sx セッション変更要求を PFCP プロトコルに送信します。
4	PFCP プロトコルが、Request-Accepted を原因として Sx セッション変更応答を SGW サービスポッドに送信します。
5	SGW サービスポッドが S5 セッション削除要求を GTPC-EP に送信します。
6	GTPC-EP がセッション削除要求を PGW に送信します。
7	PGW が、Request-Accepted を原因としてセッション削除応答を GTPC-EP に送信します。
8	GTPC-EP が、Request-Accepted を原因として S5 セッション削除応答を SGW サービスポッドに送信します。
9	SGW サービスポッドが Sx セッション削除要求を PFCP プロトコルに送信します。
10	SGW サービスポッドが、Request-Accepted を原因として PFCP プロトコルから Sx セッション削除応答を受信します。
11	SGW サービスポッドが CDR の生成をトリガーします。
12	SGW サービスポッドが CDR 要求を GTPP-EP に送信します。
13	SGW サービスポッドが、Request-Accepted を原因としてセッション削除応答を GTPC-EP に送信します。
14	GTPC-EP ポッドが、Request-Accepted を原因として MME にセッション削除応答を転送します。
15	GTPP-EP が CDR ACK を SGW サービスポッドに送信します。

しきい値に達した使用状況レポートのコールフロー

ここでは、しきい値に達した使用状況レポートのコールフローについて説明します。

図 *115 :* しきい値に達した使用状況レポートのコールフロー

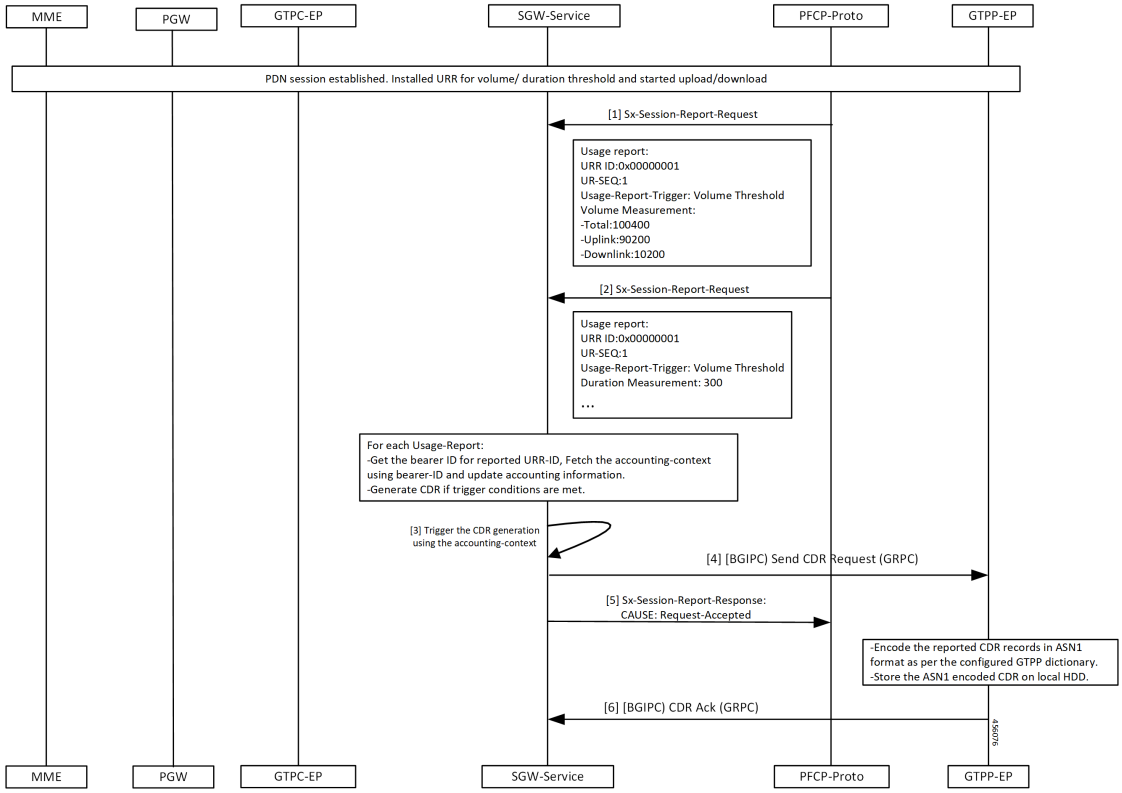


表 *220 :* しきい値に達した使用状況レポートのコールフローの説明

ステップ	説明
1	PDN セッションが確立されました。 しきい値期間に URR がインストールされました。 アップロードとダウンロードがトリガーされます。 PFCP プロトコルが Sx Session Report Request を SGW サービスポッドに送信します。
2	PFCP プロトコルが、ユーザープレーンのしきい値に達するまで、Sx Session Report Request を SGW サービスポッドに送信します。
3	SGW サービスポッドが CDR の生成をトリガーします。
4	SGW サービスポッドが CDR 要求を GTPP-EP に送信します。
5	SGW サービスポッドが Request-Accepted の理由で Sx Session Report Response を PFCP プロトコルに送信します。
6	GTPP-EP が CDR ACK を SGW サービスポッドに送信します。

専用ベアラーの URR インストールのコールフロー

ここでは、専用ベアラーの URR インストールのコールフローについて説明します。

図 *116 :* 専用ベアラーの *URR* インストールのコールフロー

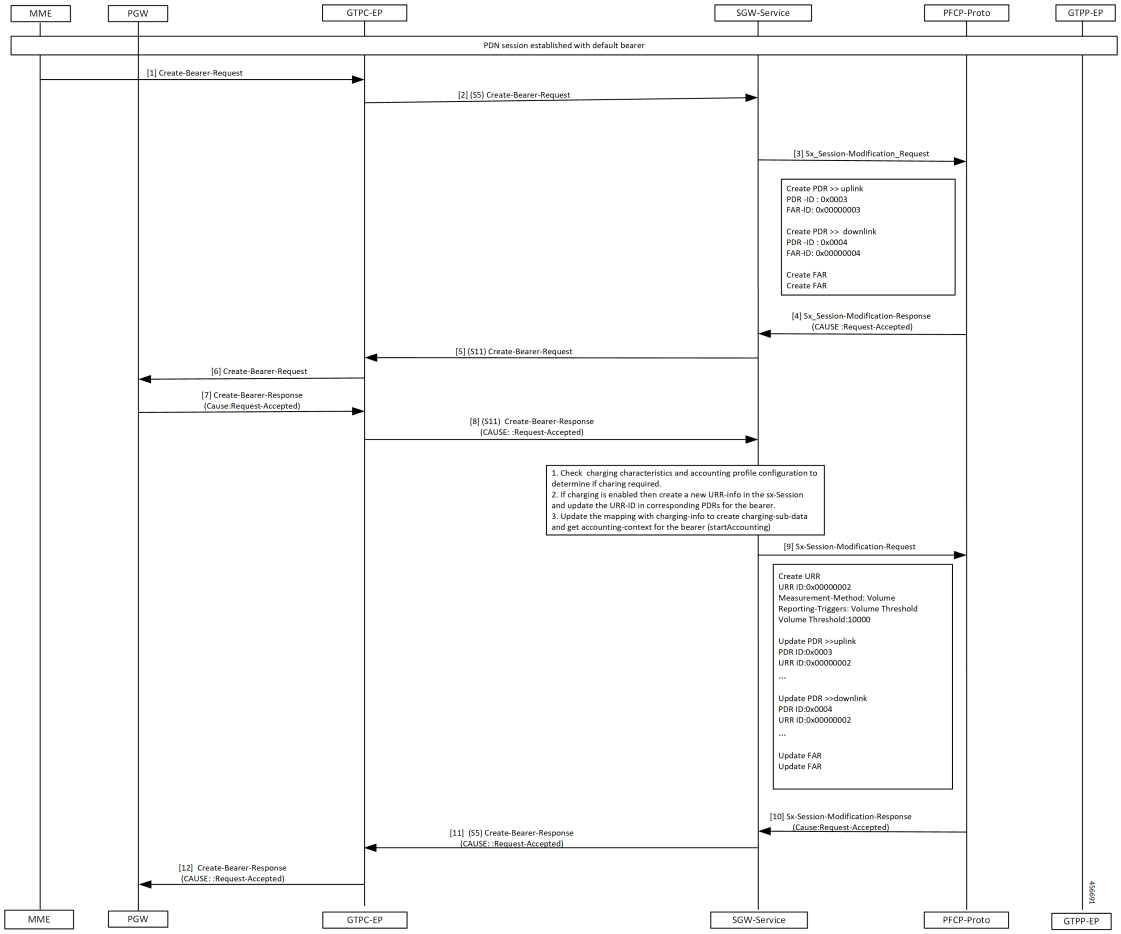


表 *221 :* 専用ベアラーの *URR* インストールのコールフローの説明

ステップ	説明
1	デフォルトベアラーで PDN セッションを確立しました。 PGW が GTPC-EP にベアラー作成要求を送信します。
2	GTPC-EP が S5 ベアラー作成要求を SGW サービスポッドに転送します。
3	SGW サービスポッドが Sx セッション変更要求を PFCP プロトコルに送信します。
4	PFCP プロトコルが、Request-Accepted を原因として Sx セッション変更応答を SGW サービスポッドに送信します。

ステップ	説明
5	SGW サービスポッドが GTPC-EP に S11 ベアラー作成要求を送信します。
6	GTPC-EP が S11 ベアラー作成要求を MME に転送します。
7	GTPC-EP が、Request-Accepted を原因として GTPC-EP に対するベアラー作成応答を受信します。
8	GTPC-EP が、Request-Accepted を原因として S11 ベアラー作成応答を SGW サービスに転送します。
9	SGW サービスポッドが Sx セッション変更要求を PFCP プロトコルに送信します。
10	PFCP プロトコルが、Request-Accepted を原因として Sx セッション変更応答を SGW サービスポッドに送信します。
11	SGW サービスポッドが、Request-Accepted を原因として S5 ベアラー作成応答を GTPC-EP に送信します。
12	GTPC-EP が、Request-Accepted を原因として PGW にベアラー作成応答を送信します。

専用ベアラーの削除時の URR の削除と CDR の生成コールフロー

ここでは、専用ベアラーの削除時の URR の削除と CDR の生成コールフローについて説明します。

図 *117 :* 専用ベアラーの削除時の *URR* の削除と *CDR* の生成コールフロー

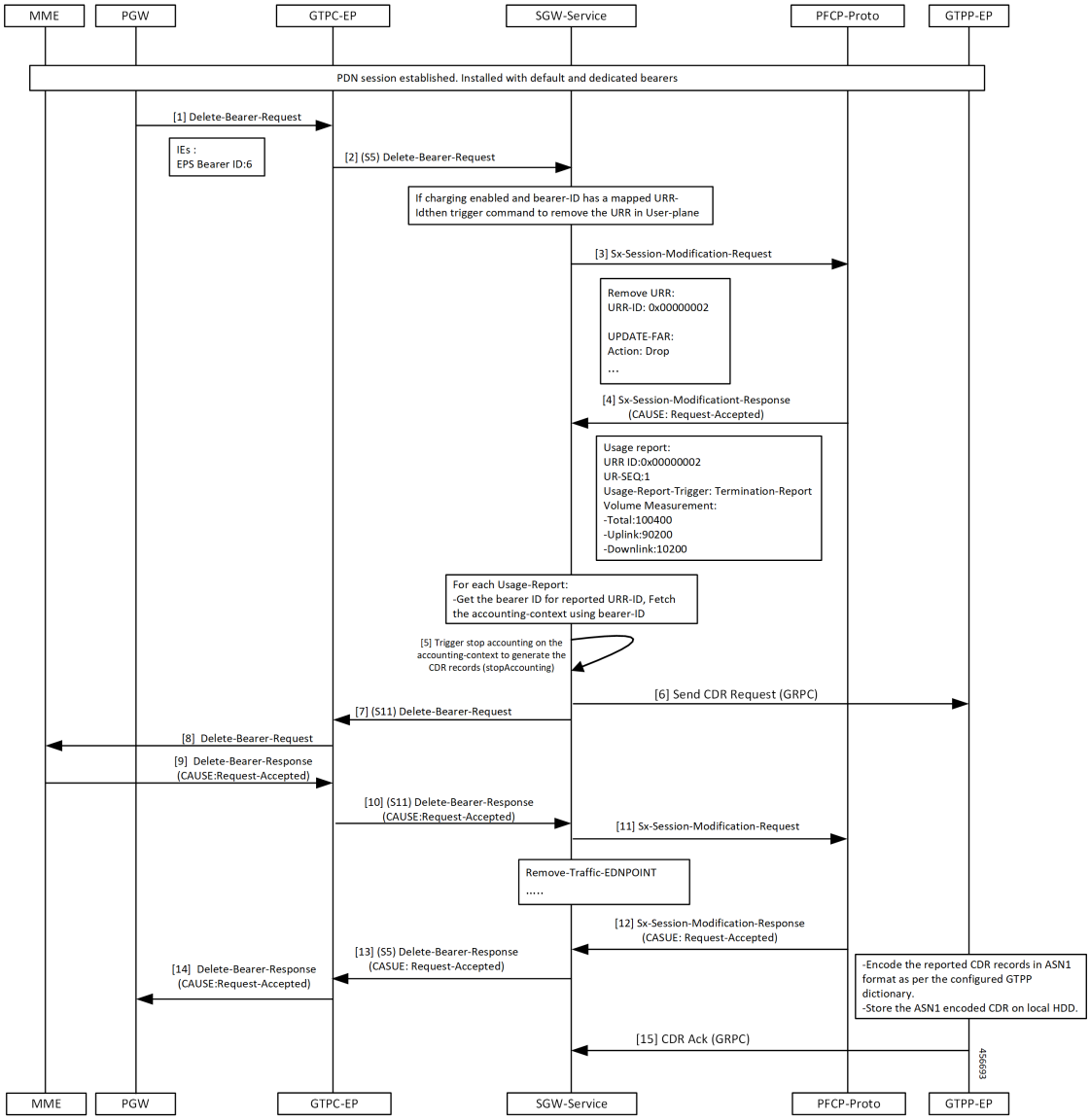


表 *222 :* 専用ベアラーの削除時の *URR* の削除と *CDR* の生成コールフローの説明

ステップ	説明
1	デフォルトのベアラーと専用ベアラーで PDN セッションが確立されました。 PGW が GTPC-EP にベアラー削除要求を送信します。
2	GTPC-EPが S5 ベアラー削除要求を SGW サービスポッドに転送します。

ステップ	説明
3	QoS トリガーと QoS の変更が検出され、課金が有効になっている場合に、SGW サービスポッドが使用状況レポートのクエリを要求します。 SGW サービスポッドが Sx セッション変更要求を PFCP プロトコルに送信します。
4	PFCP プロトコルが、Request-Accepted を原因として Sx セッション変更応答を SGW サービスポッドに送信します。
5	SGW サービスポッドが CDR の生成をトリガーし、CDR 要求を GTPP-EP に送信します。
6	SGW サービスポッドが GTPP-EP に S5 Sx ベアラー変更要求を送信します。
7	SGW サービスポッドが GTPC-EP に S11 ベアラー削除要求を送信します。
8	GTPC-EP が MME にベアラー削除要求を転送します。
9	GTPC-EP が、Request-Accepted を原因として MME からベアラー削除応答を受信します。
10	GTPC-EP が、Request-Accepted を原因として S11 ベアラー削除応答を SGW サービスポッドに転送します。
11	SGW サービスポッドが Sx セッション変更要求を PFCP プロトコルに送信します。
12	SGW サービスポッドが、Request-Accepted を原因として PFCP プロトコルから Sx セッション変更応答を受信します。
13	SGW サービスポッドが、Request-Accepted を原因として S5 ベアラー削除応答を GTPC-EP に送信します。
14	GTPC-EP が、Request-Accepted を原因として PGW にベアラー削除応答を送信します。
15	PFCP プロトコルが SGW サービスポッドに CDR ACK を送信します。

S11 トリガーに関するボリュームレポートのコールフロー

ここでは、S11 トリガーに関するボリュームレポートのコールフローについて説明します。

図 *118 : S11* トリガーに関するボリュームレポートのコールフロー

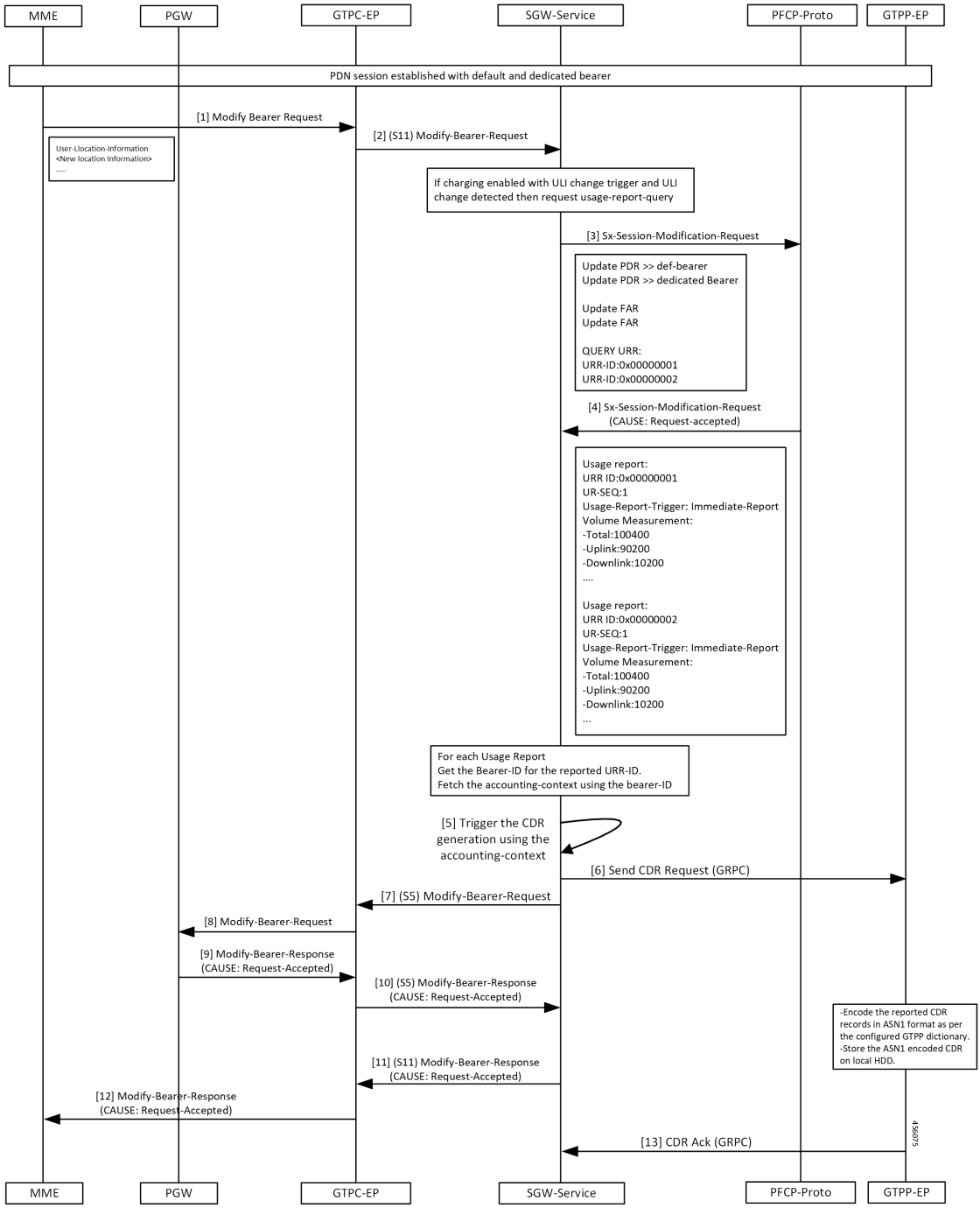


表 *223 : S11* トリガーに関するボリュームレポートのコールフローの説明

ステップ	説明
1	デフォルトのベアラーと専用ベアラーで PDN セッションが確立されました。 MME が Modify Bearer Request を GTPC-EP に送信します。
2	GTPC-EP が S11 Modify Bearer Request を SGW サービスポッドに転送します。
3	QoS トリガーと QoS の変更が検出され、課金が有効になっている場合、SGW サービスポッドが使用状況レポートのクエリを要求します。 SGW サービスポッドが Sx Session Modification Request を PFCP プロトコルに送信します。
4	PFCP プロトコルが Request-Accepted の理由で Sx Session Modification Response を SGW サービスポッドに送信します。
5	SGW サービスポッドが CDR の生成をトリガーします。
6	SGW サービスポッドが、生成された CDR 要求を GTPP-EP に送信します。
7	SGW サービスポッドが S5 Sx Modify Bearer Request を GTPC-EP に送信します。
8	GTPC-EP が Modify Bearer Request を PGW に送信します。
9	GTPC-EP が Request-Accepted の理由で PGW から Modify Bearer Response を受信します。
10	GTPC-EP が Request-Accepted の理由で S5 Modify Bearer Response を SGW サービスポッドに転送します。
11	SGW サービスポッドが Request-Accepted の理由で S11 Modify Bearer Response を PGW に送信します。
12	GTPC-EP が Request-Accepted の理由で Modify Bearer Response を MME に送信します。
13	GTPP-EP が CDR ACK を SGW サービスポッドに送信します。

S5 トリガーに関するボリューム レポート コールフロー

ここでは、S5 トリガーに関するボリューム レポート コールフローについて説明します。

図 *119 : S5* トリガーに関するボリューム レポート コールフロー

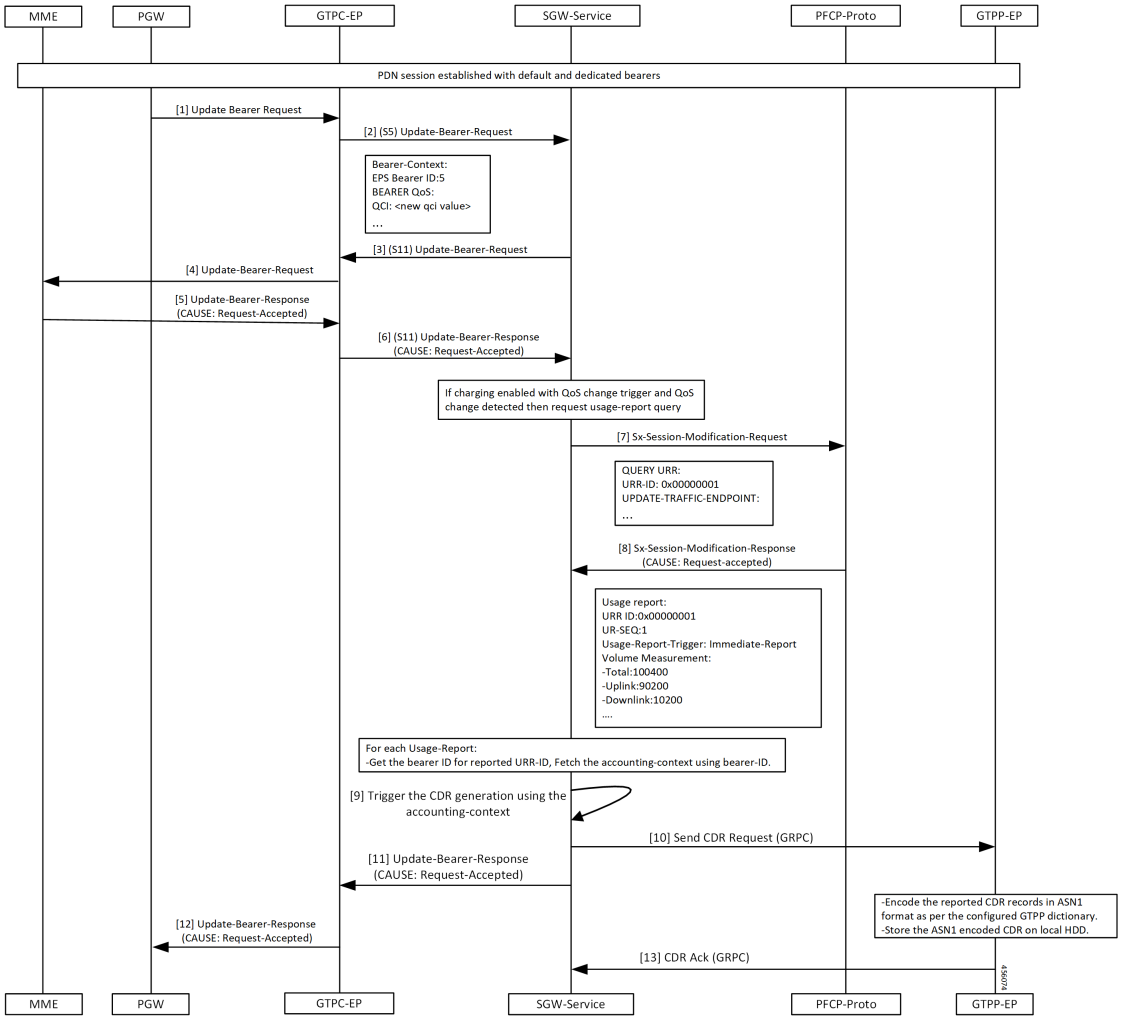


表 *224 : S5* トリガーに関するボリューム レポート コールフローの説明

ステップ	説明
1	デフォルトのベアラーと専用ベアラーで PDN セッションが確立されました。 PGW がベアラー更新要求を GTPC-EP に送信します。
2	GTPC-EP は、S5 ベアラー更新要求を SGW サービスポッドに転送します。
3	SGW がベアラー更新要求 S11 を GTPC-EP に送信します。
4	GTPC-EP がベアラー更新要求を MME に転送します。
5	MME が Request-Accepted の理由で、ベアラー更新応答を GTPC-EP に送信します。

ステップ	説明
6	GTPC-EP が Request-Accepted の理由で、S11 ベアラー更新応答を SGW サービスポッドに送信します。
7	QoS トリガーと QoS の変更が検出され、課金が有効になっている場合に、SGW サービスポッドが使用状況レポートのクエリを要求します。 SGW サービスポッドが Sx セッション変更要求を PFCP プロトコルに送信します。
8	SGW サービスポッドが、Request-Accepted の理由で PFCP プロトコルから Sx セッション変更応答を受信します。
9	SGW サービスポッドが CDR の生成をトリガーします。
10	SGW サービスポッドが CDR レポートを GTPP-EP に送信します。
11	GTPC-EP が Request-Accepted の理由で、ベアラー更新応答を SGW サービスポッドから受信します。
12	PGW が Request-Accepted の理由で、GTPC-EP からの S5 ベアラー更新応答を転送します。
13	GTPP-EP が CDR ACK を SGW サービスポッドに送信します。

標準準拠

SGW 課金サポートは、次の 3GPP 標準規格に準拠しています。

- *3GPP TS 32.251*「*Telecommunication management;Charging management;Packet Switched (PS) domainCharging*」
- *3GPP TS 32.295*「*Telecommunication management; Charging management; Charging Data Record (CDR) transfer*」
- *3GPP TS 32.297*「*Telecommunication management; Charging management; Charging Data Record (CDR) file format and transfer*」
- *3GPP TS 32.298*「*Telecommunication management; Charging management; Charging Data Record (CDR) parameter description*」

制限事項

この機能には、本リリースにおいて次の制限があります。

- 2021.02.0 リリースでは、cnSGW-C は次をサポートします。

- anp-mbr および node-id-prefix CDR 属性の有効化または無効化。他の cnSGW-C CDR 属性はデフォルトで有効になっています
- リモート SFTP サーバーへのプッシュ操作を実行する際は、暗号化された URL の構成のみ

- 2021.02.0 リリースでは、cnSGW-C は以下をサポートしていません。
  - モニタープロトコルによる CDR のサポートなし
  - デュアルスタック（IPv4v6）コールの Served PDP または PDN アドレス拡張 CDR 属性
  - 動作ビット。デフォルト値はゼロです
  - CDR ファイルの圧縮
  - ユーザーが指定した正規表現を使用した CDR ファイルの消去

- cnSGW-C 課金プロファイルの動的構成の場合：
  - 課金プロファイルの構成を動的に削除することはできません。課金プロファイル構成を削除する前に、既存のサブスクライバをクリアする必要があります。

機能設定

この機能の設定には、次の手順が含まれます。

- CLI 構成：この構成では、cnSGW-C 課金プロファイル、モード、しきい値、およびその特性を構成するコマンドを提供します。詳細については、CLI 設定 （686 ページ） を参照してください。
- CLI の表示：この構成では、SFTP プッシュ CLI を表示するコマンドを提供します。詳細については、表示 CLI （698 ページ）を参照してください。

CLI 設定

cnSGW-C 課金 CLI 設定には、次の手順が含まれます。

- 課金プロファイルまたは GTP プライム：この構成では、cnSGW-C GTPP プロファイルを構成するコマンドが提供されます。詳細については、cnSGW-C 課金プロファイルまたは GTP プライムの構成 （687 ページ） を参照してください。
- 課金モード：この構成では、cnSGW-C 課金モードを構成するコマンドが提供されます。詳細については、課金モードの設定 （693 ページ） を参照してください。
- 課金しきい値：この構成では、cnSGW-C 課金しきい値を構成するコマンドが提供されます。詳細については、cnSGW-C 課金しきい値の設定 （693 ページ） を参照してください。

- 課金しきい値と課金プロファイルの関連付け：この構成では、cnSGW-C 課金しきい値と cnSGW-C 課金プロファイルの関連付けを構成するコマンドが提供されます。詳細については、cnSGW-C 課金しきい値と cnSGW-C 課金プロファイルの関連付けの構成 （694 ページ） を参照してください。
- コール制御プロファイル：この構成では、cnSGW-C コール制御プロファイルを構成するコマンドが提供されます。詳細については、コール制御プロファイルの構成 （696ページ）を参照してください。
- コール制御プロファイルでの課金特性：この構成では、コール制御プロファイルでcnSGW-Cの課金特性を構成するコマンドが提供されます。詳細については、呼制御プロファイルでの課金特性の設定 （697 ページ）を参照してください。

cnSGW-C 課金プロファイルまたは GTP プライムの構成

（注）

- cnSGW-C 課金は、GTP プライムの複数のレプリカをサポートします。
- プライマリ ストレージ サーバーで 4 回連続して障害が発生した場合、cnSGW-C はプライマリ ストレージ サーバーからセカンダリ ストレージ サーバーに切り替えます。セカンダリ ストレージ サーバーで障害が 4 回連続して発生した場合、またはプライマリ ストレージ サーバーからセカンダリ ストレージ サーバーへのスイッチオーバーから 30 分経過した場合（いずれか早い方）、プライマリ ストレージ サーバーに切り替わります。
- CDR ファイルストレージの割り当てサイズの 95% が超過すると、古い CDR ファイルが削除されます。

cnSGW-C 課金プロファイルまたは GTP プライムの構成には、次の手順が含まれます。

- GTPP プロファイル：この構成では、cnSGW-C GTPP プロファイルを構成するコマンドが提供されます。詳細については、GTPP プロファイルの設定 （687 ページ） を参照してください。
- 既存のエンドポイント関連の CLI：この構成では、cnSGW-C の既存のエンドポイント関連の CLI を構成するためのコマンドが提供されます。詳細については、GTPP エンドポイントの設定 （690 ページ） を参照してください。
- SGW 課金プロファイル：この構成では、cnSGW-CPP プロファイルを構成するコマンドを提供します。詳細については、SGW 課金プロファイルの設定 （691 ページ）を参照してください。

GTPP プロファイルの設定

GTPP-EP ポッドが使用するサーバーの詳細、ディクショナリ、タイムアウトなどを設定できます。

GTPP プロファイルを設定するには、次の設定を使用します。

```
config
  profile gtpp-profile profile_name gtpp
     local-storage
        file
           rotation
            volume volume_value
            cdr-count cdrcount_value
            time-interval interval_value
           exit
           name
            prefix prefix_value
            format format
            max-file-seq-num max_sequence_number
            start-file-seq-num start_sequence_number
            recover-file-seq-num { true | false }
           exit
           purge-processed-files purge-interval purgeinterval_value
          exit
          push
           encrypted-url url_name
           encrypted-secondary-url url_name
          exit
         exit
         dictionary  custom_value
         end
```

注：

- **local-storage**：ローカルストレージの詳細。
- **file**：ファイルの詳細を指定します。
- **rotation**：ファイルローテーションの詳細を指定します。
- **volume** *volume_value*：ファイルローテーション用のファイルのボリュームを MiB 単位で指定します。2 ～ 40 の範囲の整数で指定する必要があります。デフォルト値は 4 です。
- **cdr-count** *cdrcount_value*：ファイルローテーションの CDR 数を指定します。1000 ～ 65000 の範囲の整数で指定する必要があります。デフォルト値は 10000 です。
- **time-interval** *interval_value*：ファイルローテーションの時間間隔を秒単位で指定します。30 ～ 86400 の範囲の整数で指定する必要があります。デフォルト値は 3600 です。
- **prefix** *prefix_value*：使用されるファイル名のプレフィックスを指定します。プレフィックス値が指定されていない場合、設定はデフォルトのプロファイル名を使用します。
- **format** *format*：ファイル形式に関連付けられた名前形式をオーバーライドするために使用するファイル名の形式を指定します。
- **max-file-seq-num** *max_sequence_number*：ロールオーバーする最大ファイルシーケンス番号を指定します。デフォルト値は 4294967295 です。

- **start-file-seq-num** *start_sequence_number*：ロールオーバー中の開始シーケンス番号を指定します。デフォルト値は 1 です。
- **recover-file-seq-num { true | false }**：true に設定すると、アプリケーションの再起動時にファイルシーケンス番号が最後のシーケンス番号から続行されます。デフォルト値は false です。
- **purge-processed-files** ：処理済みのファイルの定期的な消去を有効にします。
- **purge-processed-files purge-interval** *purgeinterval_value*：処理済みのファイルを消去する間隔を分単位で指定します。デフォルト値は 60 です。
- **encrypted-url**：CDR ファイルをプッシュするプライマリ SFTP URL を指定します。
- **encrypted-secondary-url**：プライマリホストでプッシュが失敗した場合にプッシュするセカンダリ SFTP URL を指定します。
- **dictionary** *custom_value*：CDR の ASN.1 エンコードに使用するディクショナリを指定します。

（注）

- SFTP URL のパスは、デフォルトでは、ユーザーが URL で指定した SFTP URL のホームディレクトリへの相対パスです。

  例：`encrypted-url sftp://user:pass@example.com:2020/upload/pf1`。ファイルは `%USER_HOME/upload/pf1` にプッシュされます

  例：`encrypted-url sftp://user:pass@example.com:2020`。ファイルは `%USER_HOME` にプッシュされます

- ユーザーのホームディレクトリ外のフォルダにファイルをアップロードするには、SFTP サーバーパスの先頭に **//** を付けて絶対パスを設定します。

  例：`encrypted-url sftp://user:pass@example.com:2020//var/opt`。ファイルは絶対パス `/var/opt` にプッシュされます

  アップロードを成功させるには、SFTPユーザーがこのパスへの書き込みアクセス権を持っている必要があります。

パスワードに、許容される URL 文字セット以外の特殊文字が含まれている場合は、RFC 3986 に従ってパーセントコーディングする必要があります。たとえば、パスワード「`pass!word`」を指定する URL は、`sftp://user:pass%21word@example.com/path/to/folder` と入力します。

設定例

次に、設定例を示します。

```
config
   profile gtpp-profile pf1 gtpp
      local-storage
         file
            rotation
```

```
volume   5
cdr-count 1000
time-interval 60
exit
name
   prefix  NYPCF508
   format  .%Y-%m-%d%H-%M-%S.%4Q
   max-file-seq-num     4
   start-file-seq-num   1
   recover-file-seq-num false
   exit
purge-processed-files purge-interval 10
exit
push
   encrypted-url sample.com sftp://user:pass@example.com//var/opt
   encrypted-secondary-url sftp://user:pass@mirror.example.com//var/opt
   exit
exit
dictionary custom24
end
```

GTPP エンドポイントの設定

（注）
- GTPP-EP ポッドはこの設定を使用します。
- GTPP-EP ポッドは常にノード設定を無視します。
- **k8s single-node** が **`false`** に設定されている場合、レプリカとノードの設定に関係なく、アクティブモードまたはスタンバイモードで GTPP-EP ポッドの 2 つのレプリカが生成されます。
- **k8s single-node** が **`true`** に設定されている場合、設定済みのレプリカが影響を受けます。
- **k8s use-volume-claim** が **`true`** に設定されている場合、エンドポイント GTP Prime を使用してストレージサイズの制限が設定されます。ストレージサイズ制限のデフォルト値は 1 GB です。
- システムが稼働している場合、ストレージサイズは変更できません。

GTPP エンドポイントを設定するには、次のコマンドを使用します。

```
config
 instance instance-id instance_id
  endpoint gtpprime
     replicas replicas_count
     nodes nodes_count
     storage storage_capacity
     end
```

注：

- **replicas** *replicas_count*：ノードごとのレプリカ数を指定します。整数で指定する必要があります。

- **nodes** *nodes_count*：このプロパティは無視されるため、設定は省略できます。
- **storage** *storage_capacity*：永続ボリュームのストレージサイズを GB 単位で指定します。1 ～ 20 の範囲の整数で指定する必要があります。

（注） CLI では、システムの実行中にストレージサイズを変更できません。ストレージサイズを変更するには、まずシステムを停止します。

設定例

次に、設定例を示します。

```
config
   instance instance-id 1
      endpoint gtpprime
         replicas 1
         storage  2
         end
```

SGW 課金プロファイルの設定

このセクションでは、SGW 課金プロファイルの設定方法を説明します。

次の SGW 課金プロファイルを設定できます。

- 属性の詳細と、その CDR への追加
- CDR を生成する際のさまざまなトリガー

SGW サービスポッドは、cnSGW-C 課金でこの設定を使用します。

次のコマンドを使用して、cnSGW-C 課金プロファイルを設定します。

```
config
  profile sgw-charging-profile profile_name
    gtpp-triggers
      volume-limit { enable | disable }
      time-limit { enable | disable }
      serving-node-change-limit { enable | disable }
      serving-node-plmn-change { enable | disable }
      uli-change { enable | disable }
      qos-change { enable | disable }
      ms-timezone-change { enable | disable }
    gtpp-attributes
      apn-ambr
        include-for-all-bearers
        include-for-default-bearer
        include-for-non-gbr-bearers
      node-id-suffix suffix_value
    gtpp-profile association_profile_name
    exit
```

（注） node-id-suffix の値は、実装固有です。ただし、GTPP プロファイルの一部として設定されたプレフィックスと同じ値を指定することをお勧めします。

注：

- **apn-ambr**：CDR に APN-AMBR 値を含めます。
- **node-id-suffix** *suffix_value*：CDR の Nodeid フィールドに含めるノード ID サフィックスを指定します。
- **ms-timezone-change { enable | disable }**：CDR 生成のトリガーとして、MS タイムゾーンの変更の有効化または無効化を指定します。デフォルト値は enable です。
- **qos-change { enable | disable }**：CDR へのコンテナ追加のトリガーとして、QoS の変更の有効化または無効化を指定します。デフォルト値は enable です。
- **serving-node-change-limit { enable | disable }**：CDR 生成のトリガーとして、サービングノードの変更（アドレス）の有効化または無効化を指定します。デフォルト値は enable です。
- **serving-node-plmn-change { enable | disable }**：CDR 生成のトリガーとして、サービングノードの PLMN の変更の有効化または無効化を指定します。
- **time-limit { enable | disable }]**：CDR 生成のトリガーとして、時間制限違反の有効化または無効化を指定します。デフォルト値は enable です。
- **uli-change { enable | disable }**：CDR へのコンテナ追加のトリガーとして、ULI の変更の有効化または無効化を指定します。デフォルト値は enable です。
- volume-limit { enable | disable }：CDR 生成のトリガーとして、ボリューム制限違反の有効化または無効化を指定します。デフォルト値は enable で、CDR の NodeId フィールドに含まれています。

設定例

次に、設定例を示します。

```
config
   profile sgw-charging-profile ch1
      gtpp-triggers volume-limit enable
      gtpp-triggers time-limit  enable
      gtpp-triggers serving-node-change-limit disable
      gtpp-triggers uli-change  enable
      gtpp-triggers qos-change  disable
      gtpp-triggers ms-timezone-change disable
      gtpp-attributes apn-ambr include-for-all-bearers
      gtpp-attributes node-id-suffix test
      gtpp-profile pf1
      end
```

課金モードの設定

課金モードは、アカウンティング GTPP または none（デフォルト）の cnSGW-C サービスモードを構成します。

（注） オフライン課金は、課金モードが GTPP に設定されている場合に有効にします。

課金モードを構成するには、次の構成を使用します。

```
config
  profile sgw sgw_srv_name
    charging-mode { gtpp | none }
    sgw-charging-threshold sgw_threshold_name
    sgw-charging-profile  sgw_charging-profile_name
    end
```

注：

- **charging-mode { gtpp | none }**：cnSGW-C 課金モードを指定します。
- **sgw-charging-threshold** *sgw_threshold_name*：関連付けられた cnSGW-C 課金しきい値の名前を指定します
- **sgw-charging-profile** *sgw_charging-profile_name*：関連付けられた cnSGW-C 課金プロファイルの名前を指定します

cnSGW-C 課金しきい値の設定

cnSGW-C課金しきい値を設定すると、CC（課金特性）ごとのボリューム、期間、またはバケットに対応するしきい値や制限を設定するのに役立ちます。

cnSGW-c 課金しきい値は、2 つの方法で設定できます。

方法：1

```
config
  profile sgw-charging-threshold threshold_name
    cc profile value cc_profile_value
      volume total total_value
      buckets buckets_value
      duration duration_value
      end
```

方法：2

```
config
  profile sgw-charging-threshold threshold_name
    cc profile value cc_profile_value
      volume
        total total_value
        uplink uplink_value
        downlink downlink_value
```

```
volume total
buckets buckets_value
serving-node-changesnode_changes_value
duration duration_value
end
```

注：

- **buckets** *buckets_value*：アカウンティングレコードを閉じるまでの、QoS の変更またはその他のトリガーによるトラフィック ボリューム コンテナの変更数を指定します。指定できる範囲は 1 ～ 20 で、デフォルト値は 4 です。
- **duration** *duration_value*：アカウンティングレコードを閉じる前に必要な通常の経過時間を指定します。
- **volume total**：CC ボリュームの詳細を指定します。

設定例

次に設定例を示します。

```
config
   profile sgw-charging-threshold thre1
      cc profile value 1
         volume total 100000
         buckets  1
         duration 60
         end

config
   profile sgw-charging-threshold thre1
      cc profile value 2
         volume uplink 100000
         volume downlink 100000
         buckets  1
         serving-node-changes 4
         duration 120
         end
```

（注） **gtpp-triggersserving-node-change-limit** が有効で、SGW 課金しきい値の下で **service-node-changes** が設定されている場合、CDR は 4 度のサービングノードの変更（MME）後に生成されます。

cnSGW-C 課金しきい値と cnSGW-C 課金プロファイルの関連付けの構成

このセクションでは、SGW課金しきい値とSGW課金プロファイルの関連付けを構成する方法について説明します。

この構成により、**sgw-charging-threshold** および **sgw-charging-profiles** が SGW プロファイルに関連付けられます。

cnSGW-c 課金しきい値と cnSGW-c 課金プロファイルの関連付けの構成には、2 つの方法があります。

方法 1

cnSGW-c 課金しきい値と cnSGW-c 課金プロファイルの関連付けを構成するため、次のコマンドを使用します。

```
config
  profile sgw sgw_srv_name
    locality location_code
    fqdn dnn_name
    plmn-id
      mcc mcc_value
      mnc mnc_value
    charging-mode { gtpp | none }
    sgw-charging-profile value
    sgw-charging-threshold limit_name
    end
```

方法 2

次のコマンドを使用して、SGW課金しきい値とSGW課金プロファイルの関連付けを構成します。

```
config
  profile sgw sgw_srv_name
    sgw-charging-threshold threshold_value
    locality location_code
    fqdn dnn_name
    charging-mode mode_name
    subscriber-policy policy_name
    end
```

設定例

次に、設定例を示します。

```
config
   profile sgw sgw1
      locality LOC1
      fqdn 209.165.200.254
      allowed-nssai [ slice1 ]
      plmn-id mcc 123
      plmn-id mnc 456
      charging-mode gtpp
      sgw-charging-profile ch1
      sgw-charging-threshold limit1
      end

config
   profile sgw sgw1
      sgw-charging-threshold thre1
      locality  LOC1
      fqdn 209.165.200.254
      charging-mode none
      subscriber-policy polSub
      end
```

コール制御プロファイルの構成

コール制御プロファイルの構成は、オペレータポリシーを通して通話処理ルールを定義して適用します。

コール制御プロファイルからの課金モードの値によって、cnSGW-C プロファイルに構成された値がオーバーライドされます。

（注）
- 1 つのコール制御プロファイルが 1 つのオペレータポリシーに関連付けられます。
- これはスタンドアロン構成です

cnSGW-C コール制御プロファイルの構成には、次の手順が含まれます。

- コール制御プロファイルの作成：この構成では、cnSGW-C コール制御プロファイルの作成を構成するコマンドが提供されます。詳細については、呼制御プロファイルの作成の設定 （696 ページ） を参照してください。
- オペレータポリシーの関連付け：この構成では、cnSGW オペレータポリシーの関連付けを構成するコマンドを提供します。詳細については、オペレータポリシーの関連付けの構成 （696 ページ）を参照してください。

呼制御プロファイルの作成の設定

呼制御プロファイルの作成を設定するには、次の設定を使用します。

```
config
  policy call-control-profile call_control_profile_name
    charging-mode sgw_charging_mode
    sgw-charging-profile assocaited_sgw_charging_profile
    end
```

設定例

次に、設定例を示します。

```
config
  policy call-control-profile ccp1
    charging-mode gtpp
    sgw-charging-profile ch1
    end
```

オペレータポリシーの関連付けの構成

オペレータポリシーの関連付けを構成するには、次の構成を使用します。

```
config
  policy operator operator_name
  policy dnn  dnn_policy_name
  policy network-capability network_name
  call-control-profile value
  end
```

設定例

次に、設定例を示します。

```
config
   policy operator opPol1
   policy dnn polDnn
   policy network-capability nc1
   call-control-profile ccp1
   end
```

呼制御プロファイルでの課金特性の設定

ローカル値を定義して、課金判断の対象となる課金特性のソースを選択できます。

呼制御プロファイルで課金特性を設定するには、次の設定を使用します。

```
config
   policy call-control-profile  call_control_profile_name
      sgw-charging-profile charging_type
      charging-mode mode_type
      cc prefer preference_type
      cc local-value profile index_bit
      end
```

注：

- **cc prefer local-value** および **cc prefer hlr-hss-value** はオプションのパラメータです。
- **cc prefer { hlr-hss-value | local-value }**：課金特性で使用するプリファレンスを次の中から指定します。
  - MME を介して HLR または HSS から受信した場合、hlr-hss に設定されているプリファレンス。
  - プリファレンスが local-value に設定されている場合。次の CLI を参照してください。

    ```
    cc prefer local-value
    cc local-value profile index-bit
    ```

- **cc local-value profile** ：次のように local-value パラメータ情報を指定します。
  - index_bit のデフォルト値は 8 です。
  - 課金特性（CC）優先値が local-value に設定されている場合、課金特性のプロファイルインデックスのローカル値を設定します。

設定例

次に、設定例を示します。

```
config
   policy call-control-profile CCP
      sgw-charging-profile test
      charging-mode gtpp
      cc prefer local-value
```

```
      cc local-value profile 4
      end
config
   policy call-control-profile CCP1
      sgw-charging-profile test
      charging-mode gtpp
      cc prefer hlr-hss-value
      end
```

（注） システムのデフォルト設定値である 8 を使用します。それ以外の場合は、CSR にある値を使用します。

```
config
   policy call-control-profile CCP2
      sgw-charging-profile test
      charging-mode gtpp
      cc prefer local-value
      end
```

（注） cc プロファイルのデフォルト値は 8 です。

表示 CLI

GTPP-EP SFTP Push CLI

- **show gtpp-ep endpoints**：実行中の GTPP-EP ポッドと対応する IP のリストを表示します。
- **show gtpp-ep files endpoint** *pod-name* **profile** *gtpp-profile_name*：指定された gtpp の特定の GTPP-EP ポッドのアーカイブされたファイルを表示します。
- **cdr push endpoint** *pod-name* **profile** *gtpp-profile* **filename** *file-to-be-uploaded*：使用可能なファイルを、指定された GTPP プロファイルの特定の GTPP-EP ポッド上のアーカイブフォルダにプッシュします。

cnSGW-CDR でサポートされている CDR フィールド

次の表は、使用可能なディクショナリに存在する cnSGW-CDR フィールドのリストです。

custom24 ディクショナリ

表 *225 : custom24* ディクショナリの説明

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
レコード タイプ	0	M	SGW IP-CAN ベアラーレコード。
サービス対象の IMSI	3	M	サービス対象側の IMSI。
S-GW アドレス	4	M	使用された SGW のコントロールプレーン IP アドレス
S-GW バイナリ IPv4 アドレス	4-0	M	オクテット文字列には、GGSN サービスの Gn アドレ イナリ形式で含まれています。
S-GW バイナリ IPv6 アドレス	4-0	M	前述のフィールドに含まれるオクテット文字列には、 サービスの Gn アドレスがバイナリ形式で含まれてい
課金識別子	5	M	IP-CAN ベアラー識別子。異なるレコード内で PCN に 作成された IP-CAN ベアラーを識別するため
サービングノードアドレスのリスト	6	M	このレコードで使用された、サービスノードのコント プレーン IP アドレスの一覧（例：SGSN、MME）。
サービングノードのバイナリ IPv4 アドレス	6-0	M	前述のフィールドに含まれるオクテット文字列には、 の IPv4 アドレスが含まれています。
サービングノードのバイナリ IPv6 アドレス	6-0	M	サービスノードバイナリ IPv4 アドレスフィールドに含 オクテット文字列には、MME の IPV6 アドレスが含ま ます。
アクセスポイント名のネットワーク識別子	7	M	外部パケットデータネットワークに接続されているア ポイントの論理名（APN のネットワーク識別子部分）
PDP/PDN タイプ	8	M	このフィールドは、PDN タイプ（例：IPv4、IPv6、ま IPv4v6）を示します。

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
サービス対象の PDP/PDN アドレス	9	M	PDP コンテキストまたは PDN 接続に割り当てられている アドレス（使用可能な場合）。 PDN タイプが IPv4 の場合は IPv4、PDN タイプが IPv6 また IPv4v6 の場合は IPv6。
PDP IP アドレス	9-0	M	このフィールドには、PDP コンテキストの IP アドレスが含 れます。
PDP IPv4 アドレス	9-0-0	M	PDP IP アドレスフィールドに含まれるオクテット文字列 は、SGW によってサブスクライバに割り当てられた IPv4 ドレスがバイナリ形式で含まれます。
PDP IPv6 アドレス	9-0-0	M	PDP IP アドレスフィールドに含まれるオクテット文字列 は、SGW によってサブスクライバに割り当てられた IPv6 ドレスがバイナリコーディングで含まれます。
ダイナミックアドレスフラグ	11	O	提供される PDP/PDN アドレスが動的であるかどうかを示 ます。このアドレスは、IP-CAN ベアラーのアクティブ化 初回アタッチ（E-UTRAN または S2x 経由）、および UE 要求による PDN 接続時に割り当てられます。アドレスが静 な場合、このフィールドは表示されません。
トラフィック データ ボリュームのリスト	12	M	QCI、ARP ペアの課金条件の変更のリスト。各変更にはタ ムスタンプが付与されます。課金条件は、料金期間ごとな トラフィック量を分類します。最初およびその後に変更さ た QoS と、それに対応するデータ値も一覧に含まれます。
課金条件の変更	12-0	M	各トラフィック量コンテナには、課金条件の詳細が含まれ す。通常、QoS 変更および料金変更のために新しいコンテ が作成されます。
データ量 GPRS アップリンク	12-0-3	M	このフィールドは、トラフィック量リストの ChangeOf CharCondition 要素の一部です。これには、コンテナで指定されたタイ レーム中にアップリンク方向で受信されたオクテットの数 含まれます。新しいコンテナごとにカウンタがリセットさ 累積しません。

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
データ量 GPRS ダウンリンク	12-0-4	M	このフィールドは、トラフィック量リストの ChangeC CharCondition 要素の一部です。これには、コンテナで指定されたタ レーム中にダウンリンク方向で送信されたオクテット 含まれます。新しいコンテナごとに、カウンタがリセ れ、累積しません。
状態の変更	12-0-5	M	このフィールドは、トラフィック量リストの ChangeC CharCondition 要素の一部です。これは、ユーザープレ ら UE への変更を定義します。
時刻の変更	12-0-6	M	このフィールドは、トラフィック量リストの ChangeC CharCondition 要素の一部です。状態の変更（レコート ローズなど）が発生し、コンテナが閉じられたときの ル時刻を提供します。
ユーザーロケーション情報	12-0-8	O	このフィールドには、ユーザーのロケーション情報が ます。
EPC QoS 情報	12-0-9	O	IP-CAN ベアラー固有のコンテナの場合、このフィー はIP-CANベアラーの承認済みQoSが含まれます。各Q ペアの最初のコンテナには、このフィールドが含まれ す。次のコンテナでは、以前の変更条件が「QoS の変 場合、このフィールドが表示されます。このフィール SGW-CDR にのみ適用されます。
CP CIoT EPS 最適化インジケータ	12-0-19	O	[cPCIo TEPSOptimisation Indicator] フィールドは、コン 記録されたデータ量の転送において、コントロールプ CIoT EPS 最適化が使用されているかどうかを示します は、以前のコンテナの変更条件が「ユーザープレーン への変更」の場合にのみ、トラフィックデータコンテ まれます。なお、SGW-CDR メインレベルの [CP CIoT Optimization Indicator] フィールドには、SGW-CDR が たときの CP CIoT EPS 最適化インジケータ値が含まれ
QCI	12-9-1	M	—
アップリンク MBR	12-9-2	O	—
ダウンリンク MBR	12-9-3	O	—
アップリンク GBR	12-9-4	O	—

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
ダウンリンク GBR	12-9-5	O	—
arp	12-9-6	O	—
APN AMBR アップリンク	12-9-7	O	—
APN AMBR ダウンリンク	12-9-8	O	—
拡張最大要求帯域幅アップリンク	12-9-9	O	—
拡張最大要求帯域幅ダウンリンク	12-9-10	O	—
拡張GBRアップリンク	12-9-11	O	—
extended GBR ダウンリンク	12-9-12	0	—
拡張 APN AMBR アップリンク	12-9-13	O	—
拡張 APN AMBR ダウンリンク	12-9-14	O	—
レコードオープン時間	13	M	この S-GW で IP-CAN ベアラーがアクティブ化された時刻または後続の部分レコードにおけるレコードオープン時間示すタイムスタンプ。
持続時間	14	M	このフィールドには、レコードの期間（秒単位）が含まれす。
レコードがクローズされた原因	15	M	このフィールドには、CDRがクローズされた理由が含まれす。
診断	16	O	このフィールドは、ベアラーコンテキストが解放され、**g attribute diagnostics** が設定されている場合に、CDR に挿入れます。
gsm408cause	16-0	M	—

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
レコードシーケンス番号	17	O	部分レコードのシーケンス番号。部分レコードの場合表示されます。
ノード ID（Node ID）	18	O	レコーディングエンティティの名前。
レコードの拡張機能	19	O	レコードに対するネットワークオペレータまたは製造有の拡張機能拡張機能が使用可能な場合に適用されま
ローカルレコードのシーケンス番号	20	O	このノードによって作成された連続レコード番号。こは、すべてのCDRタイプを含めて連番で割り当てられ
APN 選択モード	21	M	APN の選択方法を示すインデックス。
サービス対象の電話番号	22	M	サブスクライバのプライマリ電話番号。
課金特性	23	M	IP-CAN ベアラに適用される課金特性。
課金特性選択モード	24	O	課金特性の選択方法に関する情報を保持します。
IMS シグナリングコンテキスト	25	O	IM-CN サブシステム シグナリング フラグが設定され場合は含まれます（「[201]」を参照）。IP-CAN ベア IMS シグナリングに使用されます。
サービングノード PLMN 識別子	27	O	このレコード中に使用されるサービングノードのPLM子（MCC および MNC）（利用可能な場合）。
サービス対象の IMEISV	29	O	使用可能な場合は、ME の IMEISV。
RAT タイプ	30	O	このフィールドは、モバイルステーションによって現されている無線アクセス技術（RAT）のタイプを示しま用可能な場合）。
MS タイムゾーン	31	O	PDN コンテキストのアクティブ化または変更手順中にが SGW に提供する可能性のあるタイムゾーン IE。

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
ユーザーロケーション情報	32	O	このフィールドには、eGTP ケースの TS 29.274 で説明されているユーザーロケーション情報が含まれます（例：CGI、SAI、RAI TAI、ECGI）。 このフィールドは、SGSN または MME によって提供されIP-CAN ベアラーのアクティブ化または変更注に SGW またPGW に転送されます。
S-GW の変更	34	O	このフィールドは、これが SGW 変更後の最初のレコードあることを示すために、SGW-CDR にのみ表示されます。の場合、TRUE (FF) に設定されます。
サービングノードのタイプ	35	M	これらのフィールドには、レコード中に接続された SGWたは PGW のコントロールプレーンにおける、1 つまたは複のサービスノードタイプが含まれます。ここにリストされているサービングノードタイプは、[Serving node Address] フルドにリストされているサービングノードアドレスに順番マッピングされます。
サービングノードタイプ列挙体	35-1	M	—
使用される P-GW アドレス	36	M	このフィールドは、コントロールプレーンの PGW IP アスです。
P-GW バイナリ IPV4 アドレス	36-0	M	このフィールドには、PGW によってサブスクライバに割てられた IPv4 アドレスがバイナリ形式で含まれます。
P-GW バイナリ IPV6 アドレス	36-0	M	このフィールドには、PGW によってサブスクライバに割てられた IPv6 アドレスがバイナリ形式で含まれます。
P-GW PLMN 識別子	37	O	—
開始時刻（Start Time）	38	O	このフィールドには、ユーザー IP-CAN セッションの開始刻が保持されます。これは、IP-CAN セッション内の最初ベアラーに対する CDR に含まれます。
停止時刻	39	O	このフィールドには、ユーザー IP-CAN セッションの停止刻が保持されます。これは、IP-CAN セッション内の最後ベアラーに対する CDR に含まれます。

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
PDN 接続識別子	40	O	このフィールドには、同じ PDN 接続に属するさまざ コードを識別するための PDN 接続（IP-CAN セッショ 別子が保持されます。
iMSI 未認証フラグ	41	O	このフィールドは、提供されたサービス対象の IMSI されていないことを示します（緊急のベアラーサービ 況）。
ユーザー CSG 情報	42	O	このフィールドには、CSG セルにアクセスするユーザ CSG 情報ステータスが含まれます。これは、PLMN 内 ID、アクセスモード、およびハイブリッドアクセスが れる場合のユーザーの CSG メンバーシップに関する指 みます。これらは、GPRS の場合は *TS 29.060*、EPC の *TS 29.274* に定義されています。
cSGId	42-0	O	CSG ID は PLMN の範囲内にある一意の識別子であり、 Subscriber Group（CSG）セルまたは CSG セルのグルー 連付けられた PLMN の CSG を識別します。
cSGAccess Mode	42-1	O	cSGAccessMode。クローズドまたはハイブリッドのい です。
cSG メンバーシップ 説明	42-2	O	このフィールドは、ユーザーの CSG メンバーシップに 情報を示します。
サービス対象の PDP/PDN アドレスの拡張	43	O	このフィールドには、デュアルスタック IPv4/IPv6 が れている場合に、PDN 接続（PDP コンテキスト、IP-C アラー）用の IPv4 アドレスが含まれます。また、IPv レスはサービス対象の PDP アドレスまたは サービス対 PDP/PDN アドレスに含まれます。
PDP IPアドレス	43-0	M	このフィールドには、PDP コンテキストの IP アドレス れます。
PDP IPV4 アドレス	43-0-0	M	このフィールドには、SGW によってサブスクライバに てられた IPv4 アドレスがバイナリ形式で含まれます。
lowAccess 優先順位表示	44	O	このフィールドは、PDN 接続がマシンタイプ通信向け 先度であるかどうかを示します。

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
ダイナミックアドレス FlagExt	47	O	このフィールドは、サービス対象の IPv4 PDP/PDN アドレが動的であるかどうかを示します。このアドレスは、IP-C ベアラーのアクティブ化、初回アタッチ（E-UTRAN また S2x 経由）、および UE の要求による PDP または PDN タ IPv4/v6 接続時に割り当てられます。IPv4 アドレスが静的 場合、このフィールドは表示されません。
s-GW iPv6 アドレス	48	O	S-GW のコントロールプレーン IPv6 アドレス（IPv4/v6 デ アルスタックの場合）。
S-GW バイナリ IPv6 アドレス	48-0	O	このフィールドには、GGSN サービスの Gn アドレスがバ ナリ形式で含まれています。
サービングノードの IPv6 アドレスのリスト	49	O	IPv4/v6デュアルスタックの場合に、このレコードで使用 たサービスノードのコントロールプレーンIPv6アドレスの 覧（例：S4-SGSN、MME）。
サービングノードのバイナリ IPv6アドレス	49-0	M	このフィールドのオクテット文字列には、MME の IPV6 ア レスが含まれます。
使用される p-GW IPv6 アドレス	50	O	このフィールドは、コントロールプレーンの PGW IPv6 ア レスです（IPv4/v6 デュアルスタックの場合）。
P-GW バイナリ IPV6 アドレス	50-0	O	このオクテット文字列には、P-GW によってサブスクライ に割り当てられたIPv6アドレスがバイナリコーディングで まれます。
最後のユーザーロケーション情報	55	O	ベアラーの非アクティブ化またはセッション解放中の UE 最後のユーザーロケーション情報を示します。
最後の MS タイムゾーン	56	O	ベアラーの非アクティブ化またはセッション解放中の UE 最新のタイムゾーンを示します。
CP CIoT EPS 最適化インジケータ	59	O	このフィールドは、PDN 接続が UE とのデータ転送中に Control Plane CIoT EPS 最適化を使用しているか（つまり、 S-GW と MME 間の S11-U を介したコントロールプレーン NAS PDU）、使用していないか（つまり、S-GW と eNB 間 S1-U を介したユーザープレーン）を示します。

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
UNI PDU CP 専用フラグ	60	O	uNIPDUCP 専用フラグフィールドは、このPDN接続に「コントロレーン専用フラグ」が適用されているかどうかを示しつまり、コントロールプレーン CIoT EPS 最適化が有合に、コントロールプレーン NAS PDU のみを使用しされているかどうかを示します。コントロールプレーEPS最適化が有効な場合でも、PDU転送用のユーザーンおよびコントロールプレーン両方の UNI（つまり、からのS1-UおよびS11-U）が許可されている場合にはフィールドはフラグ付けされません。
RAN のセカンダリ RAT 使用状況レポートのリスト	64	OC	このフィールドには、セカンダリ RAT に関して RAN 告された 1 つ以上のコンテナが含まれます。
RAN のセカンダリ RAT 使用状況レポート	64-0	M	このフィールドには、セカンダリ RAT に関して RAN 告されたコンテナが含まれます。
データ量アップリンク	64-0-1	M	このフィールドには、RANから報告された、アップリ向でのパケットデータサービスの使用中に送信されたオト数が含まれます。RANからのアップリンクデータ量とレポートはオプションです。
データ通信量ダウンリンク	64-0-2	M	このフィールドには、RANから報告されたアップリンでのパケットデータサービスの使用中に送信されたオト数が含まれます。RANからのダウンリンクデータ量とレポートはオプションです。
RAN 開始時間	64-0-3	M	このフィールドは、RANがボリュームコンテナを開くタイムスタンプです。
RAN 終了時間	64-0-4	M	このフィールドは、RANがボリュームコンテナを終了刻のタイムスタンプです。
セカンダリ RAT タイプ	64-0-5	OC	このフィールドには、セカンダリ RAT の RAT タイプれます。
UE ローカル IP ポート情報	253	O	このフィールドには、S2b ユーザーのローカル IP ポーが含まれます。
UE ローカル IP アドレス	253-0	O	このフィールドには、UWAN ユーザー IP アドレスがます。

フィールド名	タグ番号（Tag Number）	カテゴリ	説明
UDP Source Port	253-1	O	このフィールドには、UWANユーザー送信元ポートが含ます。

（注） すべての IP アドレスはバイナリ形式でエンコーディングされます。

custom24 におけるフィールドの ASN.1 定義

次のセクションでは、custom24ディクショナリのすべてのcnSGW-CDR関連フィールドの完全な ASN.1 定義を示します。

```
GPRS-SGW-Charging-DataTypes-REL8 DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-------------------------------------------------------------------------------
--
--    GPRS RECORDS
--
-------------------------------------------------------------------------------

GPRSRecord::= CHOICE
--
-- Record values 20, 22..27 are specific
-- Record values 76..77 are MBMS specific
-- Record values 78..79 are EPC specific
{
    sGWRecord[78] SGWRecord
}

SGWRecord    ::= SET
{
    recordType                                [0] RecordType,
    servedIMSI                                [3] IMSI,
    s-GWAddress                               [4] GSNAddress,
    chargingID                                [5] ChargingID,
    servingNodeAddress                        [6] SEQUENCE OF GSNAddress,
    accessPointNameNI                         [7] AccessPointNameNI OPTIONAL,
    pdpPDNType                                [8] PDPType OPTIONAL,
    servedPDPPDNAddress                       [9] PDPAddress OPTIONAL,
    dynamicAddressFlag                        [11] DynamicAddressFlag OPTIONAL,
    listOfTrafficVolumes                      [12] SEQUENCE OF ChangeOfCharCondition
OPTIONAL,
    recordOpeningTime                         [13] TimeStamp,
    duration                                  [14] CallDuration,
    causeForRecClosing                        [15] CauseForRecClosing,
    diagnostics                               [16] Diagnostics OPTIONAL,
    recordSequenceNumber                      [17] INTEGER OPTIONAL,
    nodeID                                    [18] NodeID OPTIONAL,
    recordExtensions                          [19] ManagementExtensions OPTIONAL,
    localSequenceNumber                       [20] LocalSequenceNumber OPTIONAL,
    apnSelectionMode                          [21] APNSelectionMode OPTIONAL,
    servedMSISDN                              [22] MSISDN OPTIONAL,
    chargingCharacteristics                   [23] ChargingCharacteristics,
```

```
    chChSelectionMode                          [24] ChChSelectionMode OPTIONAL,
    iMSsignalingContext                        [25] NULL OPTIONAL,
    servingNodePLMNIdentifier                  [27] PLMN-Id OPTIONAL,
    servedIMEISV                               [29] IMEI OPTIONAL,
    rATType                                    [30] RATType OPTIONAL,
    mSTimeZone                                 [31] MSTimeZone OPTIONAL,
    userLocationInformation                    [32] OCTET STRING OPTIONAL,
    sGWChange                                  [34] SGWChange OPTIONAL,
    servingNodeType                            [35] SEQUENCE OF ServingNodeType,
    p-GWAddressUsed                            [36] GSNAddress OPTIONAL,
    p-GWPLMNIdentifier                         [37] PLMN-Id OPTIONAL,
    startTime                                  [38] TimeStamp OPTIONAL,
    stopTime                                   [39] TimeStamp OPTIONAL,
    pDNConnectionID                            [40] ChargingID OPTIONAL,
    servedPDPPDNAddressExt                     [43] PDPAddress OPTIONAL,
    lowAccessPriorityIndicator                 [44] NULL OPTIONAL,
    dynamicAddressFlagExt                      [47] DynamicAddressFlag OPTIONAL,
    s-GWiPv6Address                            [48] GSNAddress OPTIONAL,
    servingNodeiPv6Address                     [49] SEQUENCE OF GSNAddress OPTIONAL,
    p-GWiPv6AddressUsed                        [50] GSNAddress OPTIONAL,
    lastUserLocationInformation                [55] OCTET STRING OPTIONAL,
    lastMSTimeZone                             [56] MSTimeZone OPTIONAL,
    cPCIoTEPSOptimisationIndicator             [59] BOOLEAN OPTIONAL,
    uNIPDUCPOnlyFlag                           [60] BOOLEAN OPTIONAL,
   listOfRANSecondaryRATUsageReports         [64] SEQUENCE OF RANSecondaryRATUsageReport
 OPTIONAL,
    uELocalIPAddressPort                       [253] SEQUENCE OF UELocalIPPortInfo
OPTIONAL

}

AccessPointNameNI::= IA5String (SIZE(1..63))
 --
 -- Network Identifier part of APN in dot representation.
 -- For example, if the complete APN is 'apn1a.apn1b.apn1c.mnc022.mcc111.gprs'
 -- NI is 'apn1a.apn1b.apn1c' and is presented in this form in the CDR.

APNSelectionMode::= ENUMERATED
{
 --
 -- See Information Elements TS 29.060, TS 29.274    or TS 29.275
 --
     mSorNetworkProvidedSubscriptionVerified    (0),
     mSProvidedSubscriptionNotVerified          (1),
     networkProvidedSubscriptionNotVerified     (2)
}

CallDuration::= INTEGER
                --
                -- The call duration is counted in seconds.
                -- For successful calls /sessions / PDP contexts, this is the chargeable
 duration.
                -- For call attempts this is the call holding time.
                --


CauseForRecClosing ::= INTEGER
{
 --
 -- In PGW-CDR and SGW-CDR the value servingNodeChange is used for partial record
 -- generation due to Serving Node Address list Overflow
 -- In SGSN servingNodeChange indicates the SGSN change
 --
 -- LCS related causes belong to the MAP error causes acc. TS 29.002
```

```
 --
 -- cause codes 0 to 15 are defined 'CauseForTerm' (cause for termination)
 -- All cause values are not relevent to SGW. Refer the spec to find out the
          -- cause values for SGW.
     normalRelease                           (0),
     abnormalRelease                         (4),
     cAMELInitCallRelease                    (5),
     volumeLimit                             (16),
     timeLimit                               (17),
     servingNodeChange                       (18),
     maxChangeCond                           (19),
     managementIntervention                  (20),
     intraSGSNIntersystemChange              (21),
     rATChange                               (22),
     mSTimeZoneChange                        (23),
     sGSNPLMNIDChange                        (24),
     unauthorizedRequestingNetwork           (52),
     unauthorizedLCSClient                   (53),
     positionMethodFailure                   (54),
     unknownOrUnreachableLCSClient           (58),
     listofDownstreamNodeChange              (59)
}


ChangeCondition ::= ENUMERATED
{
     qoSChange                               (0),
     tariffTime                              (1),
     recordClosure                           (2),
     cGI-SAICHange                          (6),    -- bearer modification. CGI-SAI Change

     rAIChange                               (7),    -- bearer modification. RAI Change
     dT-Establishment                        (8),
     dT-Removal                              (9),
     eCGIChange                              (10),    -- bearer modification. ECGI Change
     tAIChange                               (11),    -- bearer modification. TAI Change
     apnAmbrChange                           (50)     -- apn-ambr change

}


ChangeOfCharCondition ::= SEQUENCE
{
        --
        -- qosRequested and qosNegotiated are used in S-CDR only
        -- ePCQoSInformation used in SGW-CDR,PGW-CDR, IPE-CDR, TWAG-CDR and ePDG-CDR
only
        -- userLocationInformation is used only in S-CDR, SGW-CDR and PGW-CDR
        -- chargingID used in PGW-CDR only when Charging per IP-CAN session is active
        -- accessAvailabilityChangeReason and relatedChangeOfCharCondition applicable
only in PGW-CDR
        -- cPCIoTOptimisationIndicator is used in SGW-CDR only
        --
        qosRequested                    [1] QoSInformation OPTIONAL,
        qosNegotiated                   [2] QoSInformation OPTIONAL,
        dataVolumeGPRSUplink            [3] DataVolumeGPRS OPTIONAL,
        dataVolumeGPRSDownlink          [4] DataVolumeGPRS OPTIONAL,
        changeCondition                 [5] ChangeCondition,
        changeTime                      [6] TimeStamp,
        userLocationInformation         [8] OCTET STRING OPTIONAL,
        ePCQoSInformation               [9] EPCQoSInformation OPTIONAL,
        chargingID                      [10] ChargingID OPTIONAL,
        userCSGInformation              [12] UserCSGInformation OPTIONAL,
        diagnostics                     [13] Diagnostics OPTIONAL,
```

```
        rATType                         [15] RATType OPTIONAL,
        uWANUserLocationInformation     [17] UWANUserLocationInfo OPTIONAL,
        cPCIoTEPSOptimisationIndicator [19] CPCIoTEPSOptimisationIndicator OPTIONAL


}

ChargingCharacteristics ::= OCTET STRING (SIZE(2))


ChargingID ::= INTEGER (0..4294967295)
 --
 -- Generated in P-GW, part of IP CAN bearer
 -- 0..4294967295 is equivalent to 0..2**32-1
 --

ChChSelectionMode    ::= ENUMERATED
{
     servingNodeSupplied        (0), -- For S-GW/P-GW
     subscriptionSpecific       (1), -- For SGSN only
     aPNSpecific                (2), -- For SGSN only
     homeDefault                (3), -- For SGSN, S-GW and P-GW
     roamingDefault             (4), -- For SGSN, S-GW and P-GW
     visitingDefault            (5) -- For SGSN, S-GW and P-GW
}

DataVolumeGPRS ::= INTEGER
 --
 -- The volume of data transferred in octets.
 --

DynamicAddressFlag ::= BOOLEAN


EPCQoSInformation ::= SEQUENCE
{
     --
     -- See TS 29.212 for more information
     --
     qCI                            [1] INTEGER,
     maxRequestedBandwithUL         [2] INTEGER OPTIONAL,
     maxRequestedBandwithDL         [3] INTEGER OPTIONAL,
     guaranteedBitrateUL            [4] INTEGER OPTIONAL,
     guaranteedBitrateDL            [5] INTEGER OPTIONAL,
     aRP                            [6] INTEGER OPTIONAL,
     apnAmbrUplink                  [7] INTEGER OPTIONAL,
     apnAmbrDownlink                [8] INTEGER OPTIONAL,
     extendedMaxRequestedBWUL       [9] INTEGER OPTIONAL,
     extendedMaxRequestedBWDL       [10] INTEGER OPTIONAL,
     extendedGBRUL                  [11] INTEGER OPTIONAL,
     extendedGBRDL                  [12] INTEGER OPTIONAL,
     extendedAPNAMBRUL              [13] INTEGER OPTIONAL ,
     extendedAPNAMBRDL              [14] INTEGER OPTIONAL
}

ETSIAddress::= AddressString
 --
 -- First octet for nature of address, and numbering plan indicator (3 for X.121)
 -- Other octets TBCD
 -- See TS 29.002
 --

GSNAddress::= IPAddress
```

```
MSNetworkCapability ::= OCTET STRING (SIZE(1..8))
 -- see TS 24.008

NetworkInitiatedPDPContext ::= BOOLEAN
 --
 -- Set to true if PDP context was initiated from network side
 --

NodeID ::= IA5String (SIZE(1..20))

NumberOfDPEncountered    ::= INTEGER


PDPAddress ::= CHOICE
{
     iPAddress         [0] IPAddress,
     eTSIAddress     [1] ETSIAddress
}

PDPType    ::= OCTET STRING (SIZE(2))
 --
 -- OCTET 1: PDP Type Organization
 -- OCTET 2: PDP Type Number
 -- See TS 29.060 for GTP, TS 29.274 for eGTP and TS 29.275 for PMIP
 --

PLMN-Id    ::= OCTET STRING (SIZE (3))
 --
 -- This is a 1:1 copy from the Routing Area Identity (RAI) IE specified in TS 29.060
 --            as follows:
 -- OCTET 1 of PLMN-Id = OCTET 2 of RAI
 -- OCTET 2 of PLMN-Id = OCTET 3 of RAI
 -- OCTET 3 of PLMN-Id = OCTET 4 of RAI
 --

QoSInformation ::= OCTET STRING (SIZE (4..255))
 --
 -- This    octet string
 -- is a 1:1 copy of the contents (i.e. starting with octet 5) of the "Bearer Quality
of
 -- Service" information element specified in TS 29.274
 --

RANSecondaryRATUsageReport ::= SEQUENCE
                -- ]
{
                dataVolumeUplink               [1] DataVolumeGPRS,
                dataVolumeDownlink             [2] DataVolumeGPRS,
                rANStartTime                   [3] TimeStamp,
                rANEndTime                     [4] TimeStamp,
                secondaryRATType               [5] SecondaryRATType OPTIONAL
}

SecondaryRATType ::= INTEGER
{
                reserved (0),
                nR (1) -- New Radio 5G
}

RATType ::= INTEGER (0..255)
 --
 -- This integer is 1:1 copy of the RAT type value as defined in TS 29.060 for GTP,
 -- TS 29.274 for eGTP and TS 29.275 for PMIP.
```

```
 --

UWANUserLocationInfo ::= SEQUENCE
{
     uELocalIPAddress    [0] IPAddress,
     uDPSourcePort       [1] OCTET STRING (SIZE(2)) OPTIONAL,
     sSID                [2] OCTET STRING OPTIONAL,     -- see format in IEEE Std
802.11-2012 [408]
     bSSID               [3] OCTET STRING OPTIONAL      -- see format in IEEE Std
802.11-2012 [408]
}

RecordType    ::= INTEGER
{
    -- Record values 0..17 are CS specific.
    -- The contents are defined in TS 32.250

    sGWRecord         (84)
}

ResultCode ::= INTEGER
 -- charging protocol return value, range of 4 byte (0...4294967259)
 -- see Result-Code AVP as used in 3GPP 32.299
 --


ServingNodeType ::= ENUMERATED
{
     sGSN                (0),
     pMIPSGW             (1),
     gTPSGW              (2),
     ePDG                (3),
     hSGW                (4),
     mME                 (5)
}

SGWChange ::= BOOLEAN
 --
 -- present if first record after inter S-GW change
 --

Diagnostics            ::= CHOICE
{
     gsm0408Cause                                    [0] INTEGER,
     -- See TS 24.008
     gsm0902MapErrorValue                    [1] INTEGER,
     -- Note: The value to be stored here corresponds to
     -- the local values defined in the MAP-Errors and
     -- MAP-DialogueInformation modules, for full details
     -- see TS 29.002
               itu-tQ767Cause                    [2] INTEGER,
     -- See ITU-T Q.767
            networkSpecificCause               [3] ManagementExtension,
     -- To be defined by network operator
            manufacturerSpecificCause     [4] ManagementExtension,
 -- To be defined by manufacturer
            positionMethodFailureCause [5] PositionMethodFailure-Diagnostic,
     -- see TS 29.002
            unauthorizedLCSClientCause [6] UnauthorizedLCSClient-Diagnostic
     -- see TS 29.002
}

IPAddress::= CHOICE
{
```

```
     iPBinaryAddress                         IPBinaryAddress,
     iPTextRepresentedAddress    IPTextRepresentedAddress
}

CPCIoTEPSOptimisationIndicator ::= BOOLEAN

IPBinaryAddress::= CHOICE
{
     iPBinV4Address     [0] OCTET STRING (SIZE(4)),
     iPBinV6Address     [1] OCTET STRING (SIZE(16))
}

IPTextRepresentedAddress::= CHOICE
{
 --
 -- IP address in the familiar "dot" notation
 --
     iPTextV4Address     [2] IA5String (SIZE(7..15)),
     iPTextV6Address     [3] IA5String (SIZE(15..45))
}

PositionMethodFailure-Diagnostic ::= ENUMERATED
{
                congestion                                              (0),
                insufficientResources                                   (1),
                insufficientMeasurementData                             (2),
                inconsistentMeasurementData                             (3),
                locationProcedureNotCompleted                           (4),
                locationProcedureNotSupportedByTargetMS                 (5),
                qoSNotAttainable                                        (6),
                positionMethodNotAvailableInNetwork                     (7),
                positionMethodNotAvailableInLocationArea                (8)
}

LocalSequenceNumber ::= INTEGER (0..4294967295)
 --
 -- Sequence number of the record in this node
 -- 0.. 4294967295 is equivalent to 0..2**32-1, unsigned integer in four octets


ManagementExtension ::= SEQUENCE
{
     identifier OBJECT IDENTIFIER,
     significance        [1]         BOOLEAN DEFAULT FALSE,
     information         [2]         ANY DEFINED BY identifier
}

ManagementExtensions ::= SET OF ManagementExtension

MSISDN::= ISDN-AddressString
                --
 -- See TS 23.003

MSTimeZone::= OCTET STRING (SIZE (2))
 --
 -- 1.Octet: Time Zone and 2. Octet: Daylight saving time, see TS 29.060


TimeStamp::= OCTET STRING (SIZE(9))
                --
                -- The contents of this field are a compact form of the UTCTime format
                -- containing local time plus an offset to universal time. Binary coded

               -- decimal encoding is employed for the digits to reduce the storage and
```

```
                -- transmission overhead
                -- e.g. YYMMDDhhmmssShhmm
                -- where
                -- YY     =            Year 00 to 99                    BCD encoded
                -- MM     =            Month 01 to 12                   BCD encoded
                -- DD     =            Day 01 to 31                       BCD encoded

                -- hh     =            hour 00 to 23                    BCD encoded
                -- mm     =            minute 00 to 59                BCD encoded
                -- ss     =            second 00 to 59                BCD encoded
                -- S      =            Sign 0 = "+", "-"            ASCII encoded
                -- hh     =            hour 00 to 23                    BCD encoded
                -- mm     =            minute 00 to 59                BCD encoded
                --

UELocalIPPortInfo   ::= SEQUENCE
{
    --
    -- The S2b user Local IP Port Information
    --
    uELocalIPAddress                          [0] IPAddress OPTIONAL,
    uDPSourcePort                              [1] INTEGER OPTIONAL
}

UELocalIPAddress::= IPAddress
UDPSourcePort           ::= INTEGER


UnauthorizedLCSClient-Diagnostic ::= ENUMERATED
{
                noAdditionalInformation                                         (0),
                clientNotInMSPrivacyExceptionList                               (1),
                callToClientNotSetup                                            (2),
                privacyOverrideNotApplicable                                    (3),
                disallowedByLocalRegulatoryRequirements                         (4),
                unauthorizedPrivacyClass                                        (5),
                unauthorizedCallSessionUnrelatedExternalClient                  (6),
                unauthorizedCallSessionRelatedExternalClient                    (7)
}

CSGAccessMode ::= ENUMERATED
{
                 closedMode  (0),
                 hybridMode  (1)
}
CSGId         ::= OCTET STRING (SIZE(4))
                 --
                 -- Defined in 23.003. Coded according to TS 29.060 for GTP, and in TS
29.274
                 -- for eGTP.
                 -- 24.008
                 --
UserCSGInformation         ::= SEQUENCE
{
                 cSGId                                          [0] CSGId,
                 cSGAccessMode                                  [1] CSGAccessMode,
                 cSGMembershipIndication                        [2] NULL OPTIONAL
}
TBCDSTRING ::= OCTET STRING
ISDN-AddressString ::= OCTET STRING
IMEI ::= TBCDSTRING (SIZE(8))
IMSI ::= TBCDSTRING (SIZE(3..8))
maxAddressLength    INTEGER ::= 20
```

```
AddressString ::= OCTET STRING (SIZE (1..maxAddressLength))
END
```

SGW 変更 OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

SGW 課金 CDR 統計

sgw_charging_cdr カウンタ

```
sgw_charging_cdr{action="close_final",app_name="SMF",cause="abnormalRelease",

cluster="Local",data_center="DC",event="AbnormalRelease",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 1
sgw_charging_cdr{action="close_final",app_name="SMF",cause="normalRelease",
cluster="Local",data_center="DC",event="NormalRelease",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 22
sgw_charging_cdr{action="close_final",app_name="SMF",cause="sGWChange",
cluster="Local",data_center="DC",event="SGWChange",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr{action="close_interim",app_name="SMF",cause="maxChangeCond",

cluster="Local",data_center="DC",event="QoSChange",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr{action="close_interim",app_name="SMF",cause="maxChangeCond",

cluster="Local",data_center="DC",event="ServingNodeChange",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 5
sgw_charging_cdr{action="close_interim",app_name="SMF",cause="timeLimit",
cluster="Local",data_center="DC",event="TimeLimit",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 56
sgw_charging_cdr{action="close_interim",app_name="SMF",cause="volumeLimit",
cluster="Local",data_center="DC",event="VolumeLimit",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr{action="open",app_name="SMF",cause="",cluster="Local",
data_center="DC",event="QoSChange",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr{action="open",app_name="SMF",cause="",cluster="Local",
data_center="DC",event="SGWChange",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr{action="open",app_name="SMF",cause="",cluster="Local",
data_center="DC",event="ServingNodeChange",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 5
sgw_charging_cdr{action="open",app_name="SMF",cause="",cluster="Local",
data_center="DC",event="StartAccounting",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 26
sgw_charging_cdr{action="open",app_name="SMF",cause="",cluster="Local",
data_center="DC",event="TimeLimit",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 56
```

```
sgw_charging_cdr{action="open",app_name="SMF",cause="",cluster="Local",
data_center="DC",event="VolumeLimit",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 3
```

SGW 課金 CDR コンテナの統計

sgw_charging_cdr_container カウンタ

```
sgw_charging_cdr_container{action="close",app_name="SMF",change_condition="qoSChange",

cluster="Local",data_center="DC",event="QoSChange",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 6
sgw_charging_cdr_container{action="close",app_name="SMF",change_condition="recordClosure",

cluster="Local",data_center="DC",event="AbnormalRelease",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 1
sgw_charging_cdr_container{action="close",app_name="SMF",change_condition="recordClosure",

cluster="Local",data_center="DC",event="NormalRelease",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 22
sgw_charging_cdr_container{action="close",app_name="SMF",change_condition="recordClosure",

cluster="Local",data_center="DC",event="SGWChange",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr_container{action="close",app_name="SMF",change_condition="recordClosure",

cluster="Local",data_center="DC",event="ServingNodeChange",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 5
sgw_charging_cdr_container{action="close",app_name="SMF",change_condition="recordClosure",

cluster="Local",data_center="DC",event="TimeLimit",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 56
sgw_charging_cdr_container{action="close",app_name="SMF",change_condition="recordClosure",

cluster="Local",data_center="DC",event="VolumeLimit",gr_instance_id="1",instance_id="0",
 pdn_plmn_type="visitor",pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr_container{action="open",app_name="SMF",change_condition="",cluster="Local",

data_center="DC",event="QoSChange",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 6
sgw_charging_cdr_container{action="open",app_name="SMF",change_condition="",cluster="Local",

data_center="DC",event="SGWChange",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 3
sgw_charging_cdr_container{action="open",app_name="SMF",change_condition="",cluster="Local",

data_center="DC",event="ServingNodeChange",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 5
sgw_charging_cdr_container{action="open",app_name="SMF",change_condition="",cluster="Local",

data_center="DC",event="StartAccounting",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"} 26
sgw_charging_cdr_container{action="open",app_name="SMF",change_condition="",cluster="Local",

data_center="DC",event="TimeLimit",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
```

```
 pdn_type="ipv4v6",service_name="sgw-service"} 56
sgw_charging_cdr_container{action="open",app_name="SMF",change_condition="",cluster="Local",

data_center="DC",event="VolumeLimit",gr_instance_id="1",instance_id="0",pdn_plmn_type="visitor",
 pdn_type="ipv4v6",service_name="sgw-service"}
```

SGW Sx レポートの統計

sgw_sx_session_report_stats カウンタ

```
sgw_sx_session_report_stats{app_name="SMF",cluster="Local",data_center="DC",
gr_instance_id="1",instance_id="0",service_name="sgw-service",status="success",
 sx_session_report_type="USAR"} 55
```

sgw_sx_usage_report_stats カウンタ

```
sgw_sx_usage_report_stats{app_name="SMF",cluster="Local",data_center="DC",
gr_instance_id="1",instance_id="0",service_name="sgw-service",status="success"}
 95
```

GTPP-EP 統計

gtpp_received_cdrs_total カウンタ

```
gtpp_received_cdrs_total{app_name="SMF",cluster="Local",data_center="DC",dictionary="custom24",
 gtpp_profile="pf1",instance_id="0",service_name="gtpp-ep"} 7
```

gtpp_processed_cdrs_total カウンタ

```
gtpp_processed_cdrs_total{app_name="SMF",cluster="Local",data_center="DC",dictionary="custom24",
 gtpp_profile="pf1",instance_id="0",service_name="gtpp-ep",status="success"}
7
```

gtpp_batched_cdrs_total ゲージ

```
gtpp_batched_cdrs_total{app_name="SMF",cluster="Local",data_center="DC",dictionary="custom24",

gtpp_profile="pf1",instance_id="0",service_name="gtpp-ep",status="batch_success"}
 2
```

gtpp_batch_flush_millis_total カウンタ

```
gtpp_batch_flush_millis_total{app_name="SMF",cluster="Local",data_center="DC",dictionary="custom24",

gtpp_profile="pf1",instance_id="0",service_name="gtpp-ep",status="batch_success"}
 1126.000588626
```

gtpp_batch_flush_duration_histgram_total カウンタ

```
gtpp_batch_flush_duration_histogram_total{app_name="SMF",bin=">5000ms",cluster="Local",data_center="DC",

dictionary="custom24",gtpp_profile="pf1",instance_id="0",service_name="gtpp-ep",status="batch_success"}
 6
```

gtpp_asn1field_encoding_failures_total

```
gtpp_asn1field_encoding_failures_total{app_name="SMF",cluster="Local",data_center="DC",gtpp_profile="pf1",
 dictionary="custom24",asn1_field="ServedIMSI",reason="Constraint
Violation",gr_instance_id="1",service_name="gtpp-ep"}1
```

第 45 章

SGW 再配置のサポート

- 機能の概要と変更履歴 （721 ページ）
- 機能説明 （722 ページ）
- 機能の仕組み （722 ページ）
- SGW リロケーション OAM サポート （741 ページ）

機能の概要と変更履歴

要約データ

表 *226* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *227* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

この機能は、次の手順をサポートしています。

- S1 ベースの SGW 再配置
- X2 ベースの SGW 再配置
- TAU SGW 再配置
- 5G から 4G の SGW 再配置

この機能は、初期接続または PDN 接続や SGW 再配置中に CS 要求でこの指示を受信した場合、PDN レベルで ePCO 通知フラグもサポートします。

次のシナリオでは、SGW は PGW への Modify Bearer Request をトリガーします。

- 送信元 MME が ePCO をサポートしていて、ターゲット MME がサポートしていない場合。
- 送信元 MME が ePCO をサポートしていて、送信元 MME がサポートしていない場合。

（注） 4G SGW 再割り当て Create Session Request メッセージで 5GS インターワーキング通知（5GSIWKI）を受信した場合は、SGW 再割り当てタイプを 5G として設定します。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、SGW 再配置機能の主要なコールフローについて説明します。

cnSGW-C への X2 ハンドオーバー SGW 再配置コールフロー

ここでは、cnSGW-C への X2 ハンドオーバー SGW 再配置コールフローについて説明します。

図 *120 : cnSGW-C* への *X2* ハンドオーバー *SGW* 再配置コールフロー

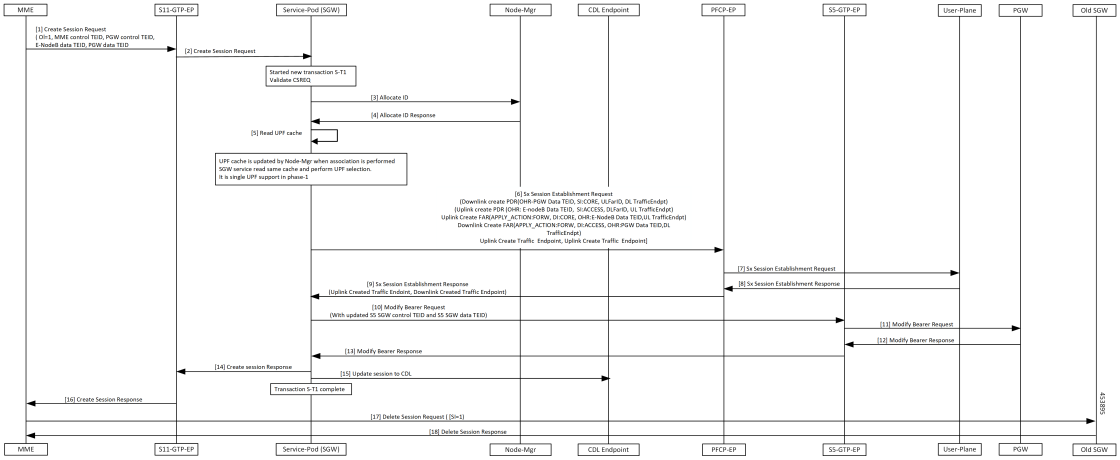


表 *228 : cnSGW-C* への *X2* ハンドオーバー *SGW* 再配置コールフローの説明

ステップ	説明
1	MME が次の情報とともに Create Session Request メッセージを GTPC-EP 入力に送信します。 • OI フラグセット • MME 制御 TEID • PGW 制御 TEID • eNodeB データ TEID • PGW データ TEID GTPC-EP 入力で新しいトランザクションを確立します。
2	SGW サービスポッドが Create Session Request を受信します。
3	SGW サービスポッド新しいトランザクション S-T1 を作成します。
4	Create Session Req を検証します。
5	ノードマネージャが TEID を割り当てます。 SGW サービスポッドが UPF キャッシュを読み取り、UPF 選択を実行します。
6	PFCP-EP が、アップリンクおよびダウンリンクの Create PDR/FAR（適用アクションは FORWARD）/CTE とともに SGW サービスポッドから Sx Session Establishment Request を受信します。
7	PFCP-EP が Sx Session Establishment Request を UPF に転送します。

ステップ	説明
8	PFCP-EP が作成済みの CTE を含む Sx Session Establishment Response を UPF から受信します。
9	SGW サービスポッドが PFCP-EP から Sx Session Establishment Response を受信します。
10	Modify Bearer Request が、SGW サービスポッドから GTPC-EP に送信された、更新済みの S5 SGW 制御 TEID および S5 SGW データ TEID で更新されます。
11	PGW が GTPC-EP から Modify Bearer Request メッセージを受信します。
12	GTCP-EP が PGW から Modify Bearer Response を受信します。
13	SGW サービスポッドが GTPC-EP から Modify Bearer Response を受信します。
14	SGW サービスポッドが Create Session Response を GTPC-EP 入力に転送します。
15	CDL でセッションが更新されます。トランザクション S-T1 が完了しました。
16	GTPC-EP 入力が Create Session Response を MME に転送します。
17	MME が SI=1 の Delete Session Request を古い SGW に送信し、Delete Session Response を受信します。 古い SGW でコールがクリアされました。

cnSGW-C への S1 ハンドオーバー SGW 再配置コールフロー

ここでは、cnSGW-C への S1 ハンドオーバー SGW 再配置コールフローについて説明します。

図 *121 : cnSGW-C* への *S1* ハンドオーバー *SGW* 再配置コールフロー

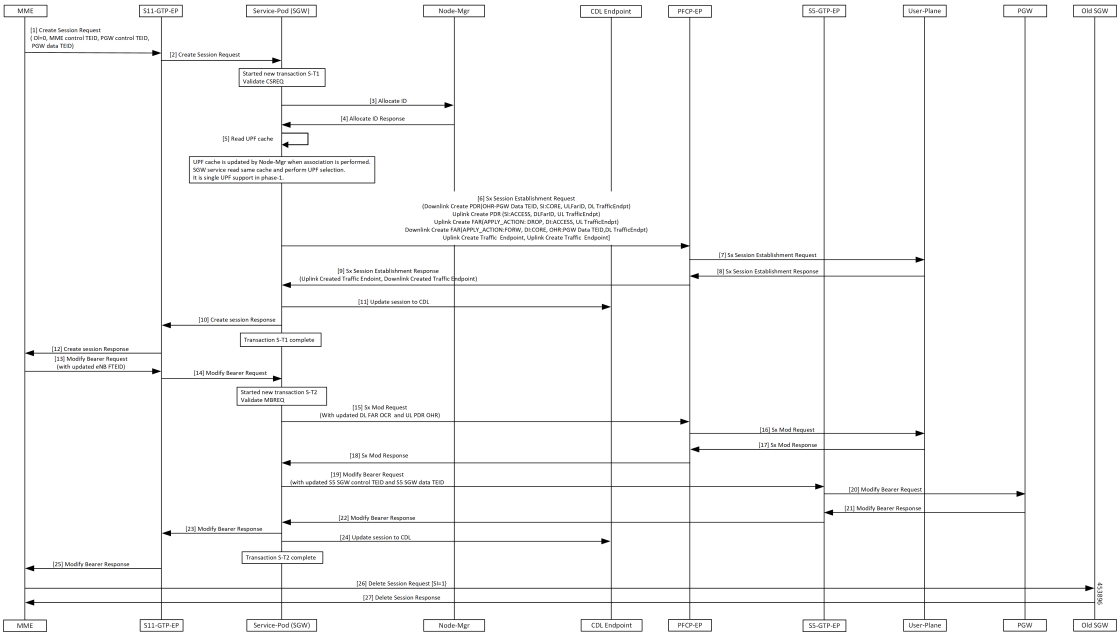


表 *229 : cnSGW-C* への *S1* ハンドオーバー *SGW* 再配置コールフローの説明

ステップ	説明
1	MME が次の情報とともにセッション作成要求メッセージを GTPC-EP 入力に送信します。 • OI フラグの設定解除 • MME 制御 TEID • PGW 制御 TEID • PGW データ TEID GTPC-EP 入力で新しいトランザクションを確立します。
2	SGW サービスポッドがセッション作成要求を受信します。
3	新しいトランザクション S-T1 を作成します。
4	セッション作成要求を検証します。
5	ノードマネージャが TEID を割り当てます。 SGW サービスが UPF キャッシュを読み取り、UPF 選択を実行します。

ステップ	説明
6	PFCP-EP が SGW サービスポッドから次の内容で Sx セッション確立要求を受信します。 • アップリンクおよびダウンリンクの PDR/FAR（Forward として ApplyAction を設定）/CTE の作成
7	PFCP-EP が Sx セッション確立要求を UPF に転送します。
8	PFCP-EP が作成済みの CTE を含む Sx セッション確立応答を UPF から受信します。
9	SGW サービスポッドが PFCP-EP から Sx セッション確立応答を受信します。
10	SGW サービスポッドが GTPC-EP 入力にセッション作成応答を転送します。
11	CDL でセッションが更新されます。トランザクション S-T1 が完了しました。
12	GTPC-EP 入力が MME にセッション作成応答を転送します。
13	GTPC-EP 入力が、更新された eNodB FTEID とともに MME からベアラー変更要求を受信します。
14	GTPC-EP 入力がベアラー変更要求を SGW サービスポッドに転送します。 新しいトランザクション T2 を作成します。
15	PFCP-EP が、更新されたダウンリンク FAR とアップリンク PDR とともに、SGW サービスポッドから Sx 変更要求を受信します。
16	PFCP-EP が Sx 変更要求を UPF に転送します。
17	PFCP-EP が UPF から Sx 変更応答を受信します。
18	SGW が PFCP-EP から Sx 変更応答を受信します。
19	SGW サービスポッドが、更新された S5 SGW 制御 TEID および S5 SGW データ TEID とともに、ベアラー変更要求を GTPC-EP に送信します。
20	GTPC-EP がベアラー変更要求を SGW に転送します。
21	GTPC-EP が PGW からベアラー変更応答を受信します。
22	SGW サービスポッドが GTPC-EP からベアラー変更応答を受信します。
23	GTPC-EP 入力が SGW サービスポッドからベアラー変更応答を受信します。
24	セッションが CDL で更新されました。トランザクション S-T2 が完了しました。
25	GTPC-EP 入力が MME にベアラー変更応答を転送します。

ステップ	説明
26	MME が SI=1 のセッション削除要求を古い SGW に送信し、セッション削除応答を受信します。 古い SGW でコールがクリアされました。

cnSGW-C への TAU X2 ハンドオーバー SGW 再配置コールフロー

このセクションでは、cnSGW-C への TAU X2 ハンドオーバー SGW 再配置コールフローについて説明します。

図 *122 : cnSGW-C* への *TAU X2* ハンドオーバー *SGW* 再配置コールフロー

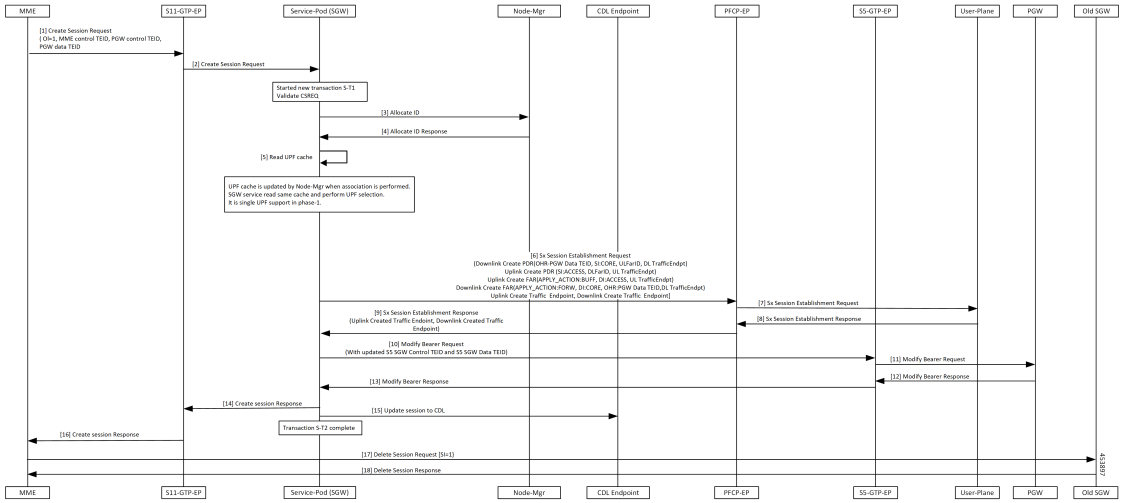


表 *230 : cnSGW-C* への *TAU X2* ハンドオーバー *SGW* 再配置コールフローの説明

ステップ	説明
1	MME が次の情報とともにセッション作成要求を GTPC-EP 入力に送信します。 • OI フラグセット • MME 制御 TEID • PGW 制御 TEID • PGW データ TEID GTPC-EP 入力で新しいトランザクションを確立します。
2	GTPC-EP 入力がセッション作成要求を SGW サービスポッドに転送します。

ステップ	説明
3	SGW サービスポッドがセッション作成要求を受信します。 新しいトランザクション S-T1 を作成します。
4	CSReq を検証します。 NodeMgr は TEID 割り当てを実行します。
5	SGW サービスが UPF キャッシュを読み取り、UPF 選択を実行します。
6	PFCP-EP が SGW サービスポッドから次の内容で Sx セッション確立要求を受信します。 • アップリンクとダウンリンクの PDR/FAR（アップリンク FAR は Forward として ApplyAction を設定）/CTE の作成。
7	PFCP-EP が Sx セッション確立要求を UPF に転送します。
8	PFCP-EP が UPF から Sx セッション確立応答を受信します。
9	SGW サービスポッドが PFCP-EP から Sx セッション確立応答を受信します。
10	SGW サービスポッドが、更新された S5 SGW 制御 TEID および S5 SGW データ TEID を使用してベアラー変更要求を GTPC-EP に送信します。
11	GTPC-EP がベアラー変更要求を SGW に転送します。
12	GTPC-EP が PGW からベアラー変更応答を受信します。
13	GTPC-EP が SGW サービスポッドにベアラー変更応答を転送します。
14	SGW サービスポッドがセッション作成要求を GTPC-EP 入力に転送します。
15	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。
16	GTPC-EP 入力が MME にセッション作成応答を転送します。
17	MME が古い SGW に送信された SI=1 のセッション削除要求を送信し、セッション削除応答を受信します。 古い SGW でコールがクリアされました。

CN-SGW（マルチ PDN）への X2 ハンドオーバー SGW 再配置のコールフロー

ここでは、cnSGW-C に向かう CN-SGW（マルチ PDN）への X2 ハンドオーバー SGW 再配置のコールフローについて説明します。

図 *123 : CN-SGW*（マルチ *PDN*）への *X2* ハンドオーバー *SGW* 再配置のコールフロー

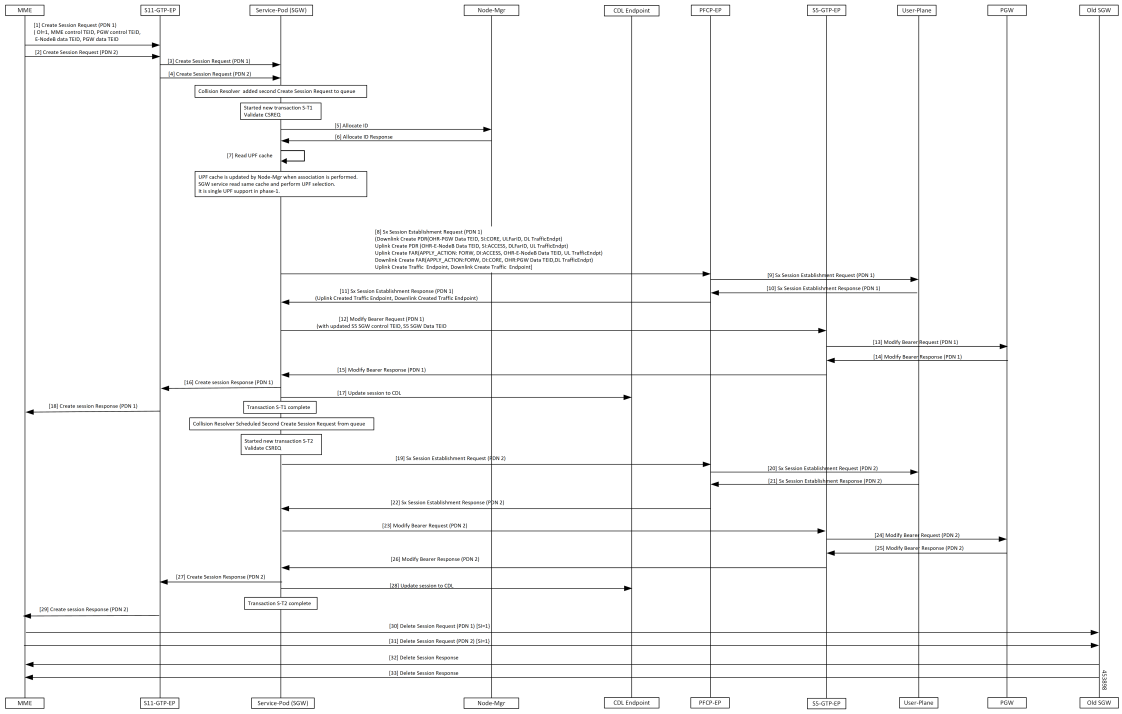


表 *231 : CN-SGW*（マルチ *PDN*）への *X2* ハンドオーバー *SGW* 再配置のコールフローの説明

ステップ	説明
1、2	MME が、GTPC-EP 入力に両方の PDN のセッション作成要求を次の内容で送信します。 • OI フラグセット • MME 制御 TEID • PGW 制御 TEID • eNodeB データ TEID • PGW データ TEID GTPC-EP 入力で新しいトランザクションを確立します。
3、4	SGW サービスポッドが PFCP-EP 入力から両方の PDN のセッション作成要求を受信します。 コリジョンリゾルバにより、PDN 2 のセッション作成要求がキューに追加されました。

ステップ	説明
5	PDN1 のセッション作成要求のために新しいトランザクション S-T1 を作成します。 CSReq を検証します。
6	ノードマネージャが TEID を割り当てます。
7	SGW サービスポッドが UPF キャッシュを読み取り、UPF 選択を実行します。
8	PFCP-EP が PDN 1 の SGW サービスポッドから次の内容で Sx セッション確立要求を受信します。 • アップリンクとダウンリンクの PDR/FAR(ApplyAction as Forward)/CTE を作成。
9	PFCP-EP が PDN 1 の Sx セッション確立要求を UPF に転送します。
10	UPF が 作成済みの CTE を含む PDN 1 の Sx セッション確立応答を PFCP-EP に送信します。
11	PFCP-EP が PDN 1 の Sx セッション確立応答を SGW サービスポッドに転送します。
12	SGW サービスポッドが GTPC-EP に PDN 1 のベアラー変更要求を次の内容で送信します。 • 更新された S5 SGW 制御 TEID • S5 SGW データ TEID
13	GTPC-EP が PGW に PDN 1 のベアラー変更要求を転送します。
14	PGW が GTPC-EP に PDN 1 のベアラー変更応答を送信します。
15	GTPC-EP が SGW サービスポッドに PDN 1 のベアラー変更応答を転送します。
16	SGW サービスが GTPC-EP 入力に PDN 1 のセッション作成応答を転送します。
17	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。 コリジョンリゾルバにより、PDN 2 のセッション作成要求がキューからスケジュールされます。
18	GTPC-EP 入力が MME に PDN 1 のセッション作成応答を転送します。
19 ～ 27	PDN2 について、8 ～ 16 の手順を繰り返します。
28	セッションが CDL で更新されました。トランザクション S-T2 が完了しました。
29	GTPC-EP 入力が MME に PDN 2 のセッション作成応答を転送します。

ステップ	説明
30 ～ 33	MME は SI=1 のセッション削除要求を古い SGW に送信し、セッション削除応答を受信します。古い SGW でコールがクリアされました。

CN-SGW（マルチ PDN）への S1 ハンドオーバー SGW 再配置コールフロー

ここでは、CN-SGW に向かう CN-SGW（マルチ PDN）への S1 ハンドオーバー SGW 再配置コールフローについて説明します。

図 *124 : CN-SGW*（マルチ *PDN*）への *S1* ハンドオーバー *SGW* 再配置コールフロー

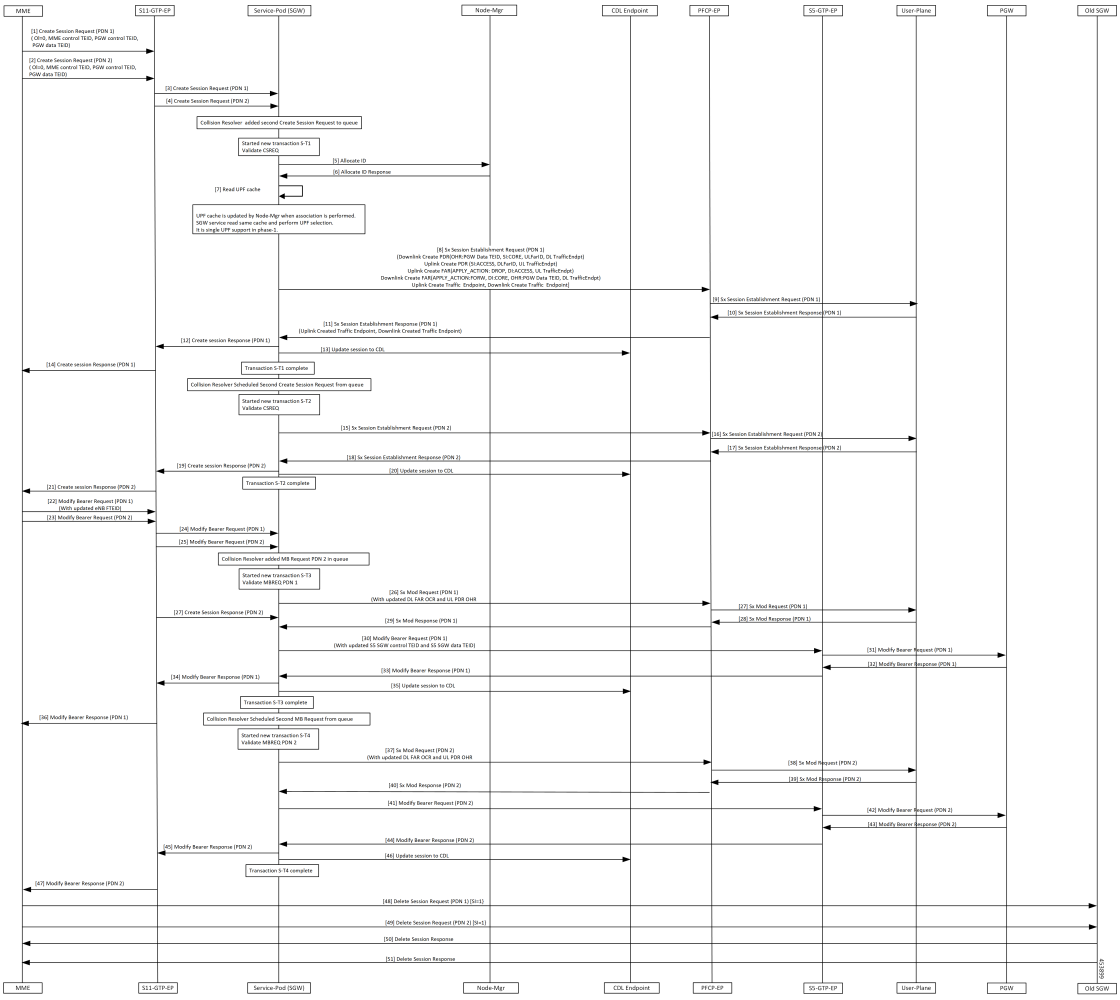


表 *232 : CN-SGW*（マルチ *PDN*）への *S1* ハンドオーバー *SGW* 再配置コールフローの説明

ステップ	説明
1、2	GTCP-EP 入力が、次の情報を含む両方の PDN のセッション作成要求を受信します。 • OI フラグの設定解除 • MME 制御 TEID • PGW 制御 TEID • PGW データ TEID GTPC-EP 入力で新しいトランザクションを確立します。
3、4	SGW サービスポッドが GTPC-EP 入力から両方の PDN のセッション作成要求を受信します。 コリジョンリゾルバにより、PDN 2 のセッション作成要求がキューに追加されました。 新しいトランザクション S-T1 を作成します。
5	CSReq を検証します。
6	ノードマネージャが TEID を割り当てます。
7	SGW サービスが UPF キャッシュを読み取り、UPF 選択を実行します。
8	PFCP-EP が PDN 1 の SGW サービスポッドから次の内容で Sx セッション確立要求を受信します。 • アップリンクとダウンリンクの PDR/FAR/CTE を作成。
9	PFCP-EP が PDN 1 の Sx セッション確立要求を UPF に転送します。
10	PFCP-PE が作成済みの CTE を含む PDN 1 の Sx セッション確立応答を UPF から受信します。
11	SGW サービスポッドが PFCP-EP から PDN 1 の Sx セッション確立応答を受信します。
12	SGW サービスポッドが GTPC-EP 入力に PDN 1 のセッション作成応答を転送します。
13	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。 コリジョンリゾルバにより、PDN 2 のセッション作成要求がキューからスケジュールされました。
14	GTPC-EP 入力が MME に PDN 1 のセッション作成応答を転送します。

ステップ	説明
15 ～ 21	PDN2（S-T2）についてステップ 11 ー 14 を繰り返します。
22、23	GTPC-EP 入力が、更新された eNodB FTEID とともに両方の PDN のベアラー変更要求を MME から受信します。
24、25	GTPC-EP が SGW サービスポッドに両方の PDN のベアラー変更応答を転送します。 コリジョンリゾルバにより、PDN2 のベアラー変更要求がキューに追加されました。 新しいトランザクション S-T3 を作成します。
26	PFCP-EP が、更新されたダウンリンク FAR とアップリンク PDR とともに、SGW サービスポッドから PDN 1 の Sx セッション変更要求を受信します。
27	PFCP-EP が PDN 1 の Sx セッション変更要求を UPF に転送します。
28	UPF が PDN 1 の Sx セッション変更応答を PFCP-EP に送信します。
29	SGW サービスポッドが PFCP-EP から PDN 1 の Sx セッション変更応答を受信します。
30	GTPC-EP が SGW サービスポッドから次の情報とともに PDN 1 のベアラー変更要求を受信します。 • 更新された S5 SGW 制御 TEID および S5 SGW データ TEID。
31	GTPC-EP が PGW に PDN 1 のベアラー変更要求を転送します。
32	GTPC-EP が PGW から PDN 1 のベアラー変更応答を受信します。
33	GTPC-EP が SGW サービスポッドに PDN 1 のベアラー変更応答を転送します。
34	SGW サービスポッドが GTPC-EP 入力に PDN 1 のベアラー変更応答を転送します。
35	セッションが CDL で更新されました。
36	GTPC-EP 入力が MME に PDN 1 のベアラー変更応答を転送します。
37	SGW サービスポッドが PFCP-EP に PDN 2 のベアラー変更要求を送信します。 トランザクション S-T3 が完了しました。 コリジョンリゾルバが、PDN 2 へのベアラー変更要求をスケジュールします。
38 ～ 40	PDN 2 について、ステップ 27、28、29 を繰り返します。
41	SGW サービスポッドが GTPC-EP に PDN 2 のベアラー変更要求を送信します。

ステップ	説明
42 ～ 47	PDN 2 について、ステップ 31 ー 36 を繰り返します。
48、49	MME が SI=1 を指定した両方の PDN のセッション削除要求を古い S-GW に送信します。
50	MME がセッション削除応答を受信します。 古い SGW でコールがクリアされました。

ベアラーコンテキストが削除対象としてマークされた状態での X2 ハンドオーバー SGW 再配置のコールフロー

このセクションでは、ベアラーコンテキストが削除対象としてマークされた状態での X2 ハンドオーバー SGW 再配置のコールフローについて説明します。

図 *125 :* ベアラーコンテキストが削除対象としてマークされた状態での *X2* ハンドオーバー *SGW* 再配置のコールフロー

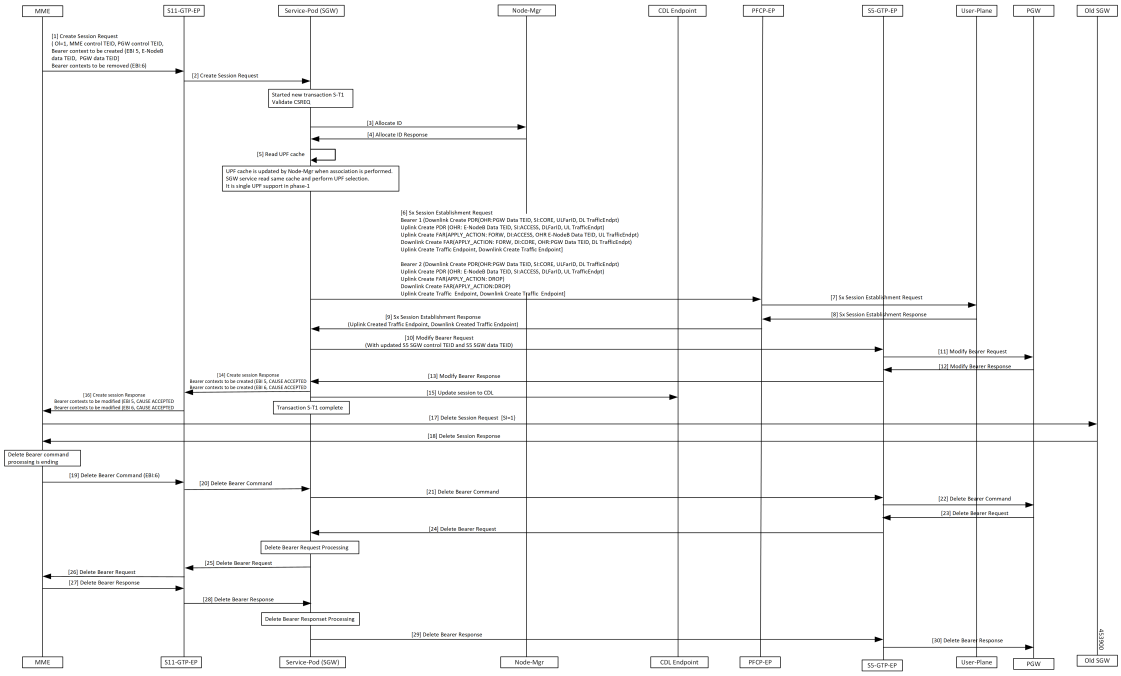


表 *233 :* ベアラーコンテキストが削除対象としてマークされた状態での *X2* ハンドオーバー *SGW* 再配置のコールフローの説明

ステップ	説明
1	GTPC-EP 入力が、次の情報を含むセッション作成要求を受信します。 • OI フラグセット • MME 制御 TEID • PGW 制御 TEID • 作成する新しいベアラーコンテキスト（EBI:5、eNodeB データ TEID、PGW データ TEID） • 削除するベアラーコンテキスト（EBI：6） GTPC-EP で新しいトランザクションを確立します。
2	GTPC-EP の入力が、セッション作成要求を SGW サービスポッドに転送します。
3	SGW サービスポッドがセッション作成要求を受信します。 新しいトランザクション S-T1 を作成します。
4	CSReq を検証します。 NodeMgr が TEID を割り当てます。
5	SGW サービスポッドが UPF キャッシュを読み取り、UPF 選択を実行します。
6	PFCP-EP が SGW サービスポッドから次の内容で Sx セッション確立要求を受信します。 • アップリンクとダウンリンクで PDR/FAR（EBI 5 は「Forward」、EBI 6 は「Drop」として ApplyAction を設定）/CTE を作成します。
7	PFCP-EP が、Sx セッション確立要求を UPF に転送します。
8	PFCP-EP が作成済みの CTE を含む Sx セッション確立応答を UPF から受信します。
9	SGW サービスポッドが PFCP-EP から Sx セッション確立応答を受信します。
10	SGW サービスポッドが、更新された S5 SGW 制御 TEID および S5 SGW データ TEID を含む変更要求を GTPC-EP に送信します。
11	GTPC-EP がベアラー変更要求を PGW に転送します。
12	GTPC-EP が PGW からベアラー変更要求を受信します。
13	GTPC-EP が SGW サービスポッドにベアラー変更応答を転送します。

ステップ	説明
14	SGW サービスポッドが両方のベアラーコンテキストに対して「Accepted」（許可）の原因を含むセッション作成要求を GTPC-EP 入力に送信します。
15	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。
16	GTPC-EP 入力がセッション作成応答を MME に転送します。
17	MME が SI=1 を指定したセッション削除要求を古い S-GW に送信します。
18	MME がセッション削除応答を受信します。古い SGW でコールがクリアされました。
19	GTPC-EP 入力が、削除対象のベアラーコンテキスト（EBI6）に対するベアラー削除コマンドを MME から受信します。
20	SGW サービスポッドが GTPC-EP 入力からベアラー削除コマンドを受信します。
21	SGW サービスポッドがベアラー削除コマンドを GTPC-EP に転送します。
22	GTPC-EP がベアラー削除コマンドを PGW に転送します。
23	PGW は GTPC-EP にベアラー削除要求（EBI 6）で応答します。
24	GTPC-EP が SGW サービスポッドにベアラー削除要求を転送します。
25	SGW サービスポッドがベアラー要求を処理し、GTPC-EP に送信します。
26	GTPC-EP 入力がベアラー削除要求を MME に転送します。
27	MME が GTPC-EP 入力にベアラー削除応答で応答します。
28	GTPC-EP 入力がベアラー削除応答をSGW サービスポッドに転送します。
29	SGW サービスポッドがベアラー削除応答を処理し、GTPC-EP に送信します。
30	GTPC-EP が PGW にベアラー削除応答を転送します。

ベアラーコンテキストが削除対象としてマークされた状態での S1 ハンドオーバー SGW 再配置のコールフロー

このセクションでは、ベアラーコンテキストが削除対象としてマークされた状態での S1 ハンドオーバー SGW 再配置のコールフローについて説明します。

図 *126 :* ベアラーコンテキストが削除対象としてマークされた状態での *S1* ハンドオーバー *SGW* 再配置のコールフロー

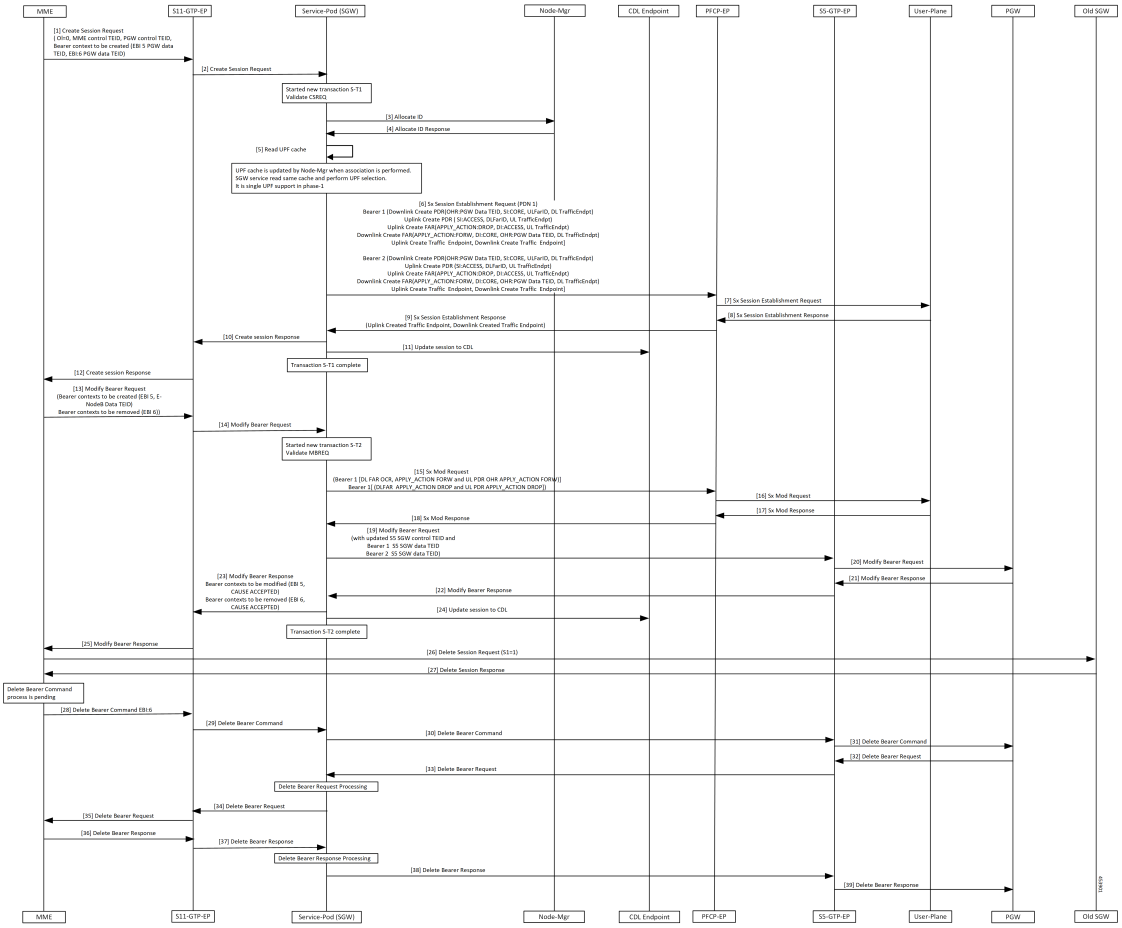


表 *234 :* ベアラーコンテキストが削除対象としてマークされた状態での *S1* ハンドオーバー *SGW* 再配置のコールフローの説明

ステップ	説明
1	GTPC-EP 入力は、MME から次の情報を含むセッション作成要求を受信します。 • OI フラグの設定解除 • MME 制御 TEID • PGW 制御 TEID • 作成するベアラーコンテキスト（EBI:5、PGW データ TEID、EBI:6、PGW データ TEID） GTPC-EP 入力で新しいトランザクションを確立します。
2	GTPC-EP の入力は、セッション作成要求を SGW サービスポッドに転送します。

ステップ	説明
3	SGW サービスポッドがセッション作成要求を受信します。 新しいトランザクション S-T1 を作成します。
4	CSReq を検証します。 NodeMgr が TEID を割り当てます。
5	SGW サービスが UPF キャッシュを読み取り、UPF 選択を実行します。
6	PFCP-EP が SGW サービスから次の内容で Sx セッション確立要求を受信します。 • アップリンクとダウンリンクで PDR/FAR（EBI 5 は「Forward」、EBI 6 は「Drop」として ApplyAction を設定）/CTE を作成します。
7	PFCP-EP が、Sx セッション確立要求を UPF に転送します。
8	PFCP-EP が、作成済みの CTE を含む Sx セッション確立応答を UPF から受信します。
9	SGW サービスが PFCP-EP から Sx セッション確立応答を受信します。
10	SGW が Sx セッション確立応答を GTPC-EP 入力に転送します。
11	セッションが CDL で更新されました。トランザクション S-T1 が完了しました。
12	GTPC-EP 入力がセッション作成応答を MME に送信します。
13	GTPC-EP 入力が、次の情報とともにベアラー変更要求を受信します。 • 新しいベアラーコンテキストを含む更新済みの eNodeB FTEID（ここでは EBI 5）と、削除されたベアラーコンテキスト（ここでは EBI 6）。
14	GTPC-EP がベアラー変更要求を SGW に転送します。
15	新しいトランザクション S-T2 を作成します。 PFCP-EP が SGW サービスポッドから次の内容で Sx セッション変更要求を受信します。 • 更新済みのダウンリンク FAR とアップリンク PDR（ベアラー 2 には DROP としてアクションを適用）。
16	PFCP-EP が Sx セッション変更要求を UPF に転送します。
17	UPF が Sx 変更応答を PFCP-EP に送信します。
18	PFCP-EP が Sx 変更応答を SGW サービスポッドに転送します。

ステップ	説明
19	SGW サービスポッドが、更新された S5 SGW 制御 TEID および S5 SGW データ TEID を含む変更要求を GTPC-EP に送信します。
20	GTPC-EP がベアラー変更要求を PGW に転送します。
21	GTPC-EP が PGW からベアラー変更要求を受信します。
22	GTPC-EP が SGW サービスポッドにベアラー変更応答を転送します。
23	SGW サービスポッドが両方のベアラーコンテキストに対して「Accepted」の原因を含むセッション作成要求を GTPC-EP 入力に転送します。
24	セッションが CDL で更新されました。トランザクション S-T2 が完了しました。
25	GTPC-EP 入力がベアラー更新応答を MME に送信します。
26	MME が SI=1 を指定したセッション削除要求を古い S-GW に送信します。
27	MME がセッション削除応答を受信します。 古い SGW でコールがクリアされました。
28	GTPC-EP 入力が、削除対象のベアラーコンテキスト（EBI 6）に対するベアラー削除コマンドを MME から受信します。
29	SGW サービスポッドが GTPC-EP 入力からベアラー削除コマンドを受信します。
30	SGW サービスポッドがベアラー削除コマンドを GTPC-EP に転送します。
31	GTPC-EP がベアラー削除コマンドを PGW に転送します。
32	PGW は GTPC-EP にベアラー削除要求（EBI 6）で応答します。
33	GTPC-EP が SGW サービスポッドにベアラー削除要求を送信します。
34	SGW サービスポッドがベアラー要求を処理し、GTPC-EP 入力に送信します。
35	GTPC-EP 入力がベアラー削除要求を MME に送信します。
36	MME が GTPC-EP 入力にベアラー削除応答で応答します。
37	GTPC-EP 入力がベアラー削除応答を受信し、SGW サービスポッドに送信します。
38	SGW サービスポッドがベアラー削除応答を処理し、GTPC-EP に転送します。
39	GTPC-EP が PGW にベアラー削除応答を転送します。

ベアラーコンテキストの数が少ない場合の MME 間および MME 内のハンドオーバーと S1 SGW 再配置コールフロー

ここでは、ベアラーコンテキストの数が少ない場合の MME 間および MME 内のハンドオーバーと S1 SGW 再配置コールフローについて説明します。

図 *127 :* ベアラーコンテキストの数が少ない場合の *MME* 間および *MME* 内のハンドオーバーと *S1 SGW* 再配置コールフロー

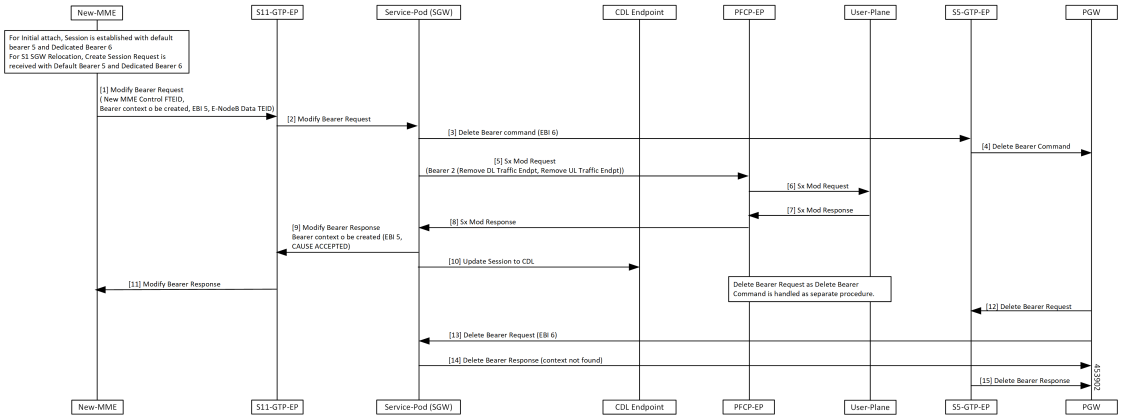


表 *235 :* ベアラーコンテキストの数が少ない場合の *MME* 間および *MME* 内のハンドオーバーと *S1 SGW* 再配置コールフローの説明

ステップ	説明
1	デフォルトのベアラー EBI 5 および専用ベアラー 6 との新しい MME セッションが確立されました。 新しい MME が、デフォルトのベアラー EBI 5 および専用ベアラー 6 を使用して、S1 SGW 再配置用の Create Session Request を受信します。 新しい MME が、次の情報とともに Modify Bearer Request を GTPC-EP 入力に送信します。 • 新しい MME 制御 TEID • 作成する新しいベアラーコンテキスト • EBI5 • eNodeB データ TEID
2	GTPC-EP 入力が Modify Bearer Request を SGW サービスポッドに転送します。
3	SGW サービスポッドが EBI 6 を使用して Delete Bearer Command を GTPC-EP に送信します。
4	GTPC-EP が Delete Bearer Command を PGW に転送します。

ステップ	説明
5	SGW サービスポッドが、ベアラー 2 の Remove DL Traffic Endpoint と Remove UL Traffic Endpoint を指定して、Sx Mod Req を PFCP-EP に送信します。
6	PFCP-EP が Sx Mod Req を UPF に転送します。
7	UPF が Sx Mod Response を PFCP-EP に送信します。
8	PFCP-EP が Sx Mod Response を SGW サービスポッドに転送します。
9	SGW サービスポッドが、次の情報とともに Modify Bearer Response を GTPC-EP 入力に送信します。 • Accepted の理由で変更するベアラーコンテキスト EBI 5
10	セッションが CDL に対して更新されました。
11	GTPC-EP 入力が Modify Bearer Response を新しい MME に送信します。
12	PGW が Delete Bearer Request を GTPC-EP に送信します。
13	SGW サービスポッドが、個別の手順として処理される Delete Bearer Command として、EBI6 を使用して PGW から Delete Bearer Request を受信します。
14	SGW サービスポッドが Context Not Found の理由で、Delete Bearer Response で GTPC-EP に応答します。
15	GTPC-EP が Delete Bearer Response を PGW に転送します。

（注）
- Sx Modify Request メッセージは、Remove DL Traffic Endpoint および Remove UL Traffic Endpoint とともに、レガシー CUPS で確認されないメッセージとして送信されます。Sx Modify for MBReq メッセージが Sx Modify Request に続き、UPF に送信されます。
- UPF は cnSGW-C で次のメッセージを受信します。
  - MB 要求の単一 Sx Modify Request メッセージ
  - Remove DL Traffic Endpoint
  - Remove UL Traffic Endpoint

SGW リロケーション OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

第 46 章

Sx 負荷/過負荷制御の処理

- 機能の概要と変更履歴 （743 ページ）
- 機能説明 （744 ページ）
- 機能の仕組み （744 ページ）
- ピア過負荷シナリオ中の Sx メッセージの処理 （745 ページ）
- 自己過負荷状態での Sx 要求メッセージのスロットリングと除外 （749 ページ）
- Sx 負荷/過負荷制御 OAM のサポート （750 ページ）

機能の概要と変更履歴

要約データ

表 *236* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	無効：有効にするには設定が必要です
関連資料	該当なし

マニュアルの変更履歴

表 *237* : マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2021.01.0

機能説明

この機能は、Sx インターフェイスを介したメッセージの負荷と過負荷制御を処理します。Sx メッセージの負荷および過負荷制御には 2 つのシナリオがあります。

- **ピアの過負荷**：ピアが過負荷状態になっている場合、負荷制御を使用すると、ユーザープレーン機能でその負荷情報をコントロールプレーン機能に送信できます。この負荷情報により、ユーザープレーン機能全体で PFCP セッションの負荷がそれぞれの有効負荷に応じて分散されます。

  過負荷機能は、特定のユーザープレーンへの新しいセッション要求のスロットリングに関する情報を制御します。UP の選択は、ユーザープレーンが LCI（負荷制御情報）と OCI（過負荷制御情報）を報告するときに実行されます。

- **自己過負荷状態**：自己保護状態時に着信メッセージを受信すると、cnSGWc は対応する DNN、ARP、QCI、メッセージの優先順位値を検証します。その結果に基づいて、cnSGWc は、過負荷除外プロファイルの設定に従って、メッセージをスロットリングするか、スロットリングから除外します。

機能の仕組み

Sx インターフェイスの負荷と過負荷の制御は、次の 2 つの部分で構成されます。

- [Peer Overload Handling]：UPF が過負荷になった場合のメッセージの処理。
- [Self Overload Handling]：cnSGWc が過負荷になった場合のメッセージの処理。

ノード機能のサポート

3GPP 標準規格に準拠：

- CP が負荷および過負荷機能をユーザープレーンに通知します。
- ユーザープレーンが CP ピアに負荷または過負荷の情報を送信するかどうかを決定します。

PFCP Sxa エンドポイントノード機能の一部として、CP で負荷および過負荷機能を設定します。この設定は、設定の変更が発生したときに、Sx 関連付け応答メッセージまたは Sx 関連付け更新要求メッセージ中に UP に伝達されます。

CP Function Features IE は、CP でサポートされている機能を示します。この IE には、（システム全体の）UP 機能の動作に影響を与える機能が含まれています。

（注） CP が CLI を介した負荷または過負荷機能をサポートしていない場合、UP 選択プロセスでは、ユーザープレーンで報告された負荷または過負荷情報は無視されます。

UP の選択

UP の選択は LCI 値ごとにのみ行われますが、スロットリングは OCI 値ごとにのみ行われます（3GPP 規格で指定）。

ピアレベルごとの LCI および OCI の表示：

```
show peers | tab | exclude rest
                                                                    POD
ENDPOINT  LOCAL ADDRESS  PEER ADDRESS  DIRECTION  INSTANCE  TYPE  CONNECTED TIME  RPC
ADDITIONAL DETAILS
--------------------------------------------------------------------------------------------------------------------------------
S5/S8  <nil>:2123  209.165.202.143:2123  Inbound  nodemgr-0  Udp  6 minutes   SGW
Recovery: 10
SXA  209.165.200.226:8805  209.165.202.143:8805  Inbound  nodemgr-0  Udp  About a minute
  SGW-U  Capacity: 65535,
LoadMetric: 20,LoadSeqNo: 1,OverloadMetric: 0,OverloadSeqNo: 0,Priority: 10
SXA  209.165.200.226:8805  209.165.202.147:8805  Inbound  nodemgr-0  Udp  2 minutes
SGW-U  Capacity: 10,
LoadMetric: 40,LoadSeqNo: 1,OverloadMetric: 100,OverloadSeqNo: 1,Priority: 20
SXA  209.165.200.226:8805  209.165.202.159:8805  Inbound  nodemgr-0  Udp  2 minutes
SGW-U  Capacity: 10,
LoadMetric: 100,LoadSeqNo: 1,OverloadMetric: 77,OverloadSeqNo: 1,Priority: 1
```

ピア過負荷シナリオ中の Sx メッセージの処理

ここでは、ピアノードが過負荷状態になっている場合の Sx メッセージの処理メカニズムについて説明します。

Sx Establishment に対するスロットリングのサポート

ユーザープレーンが過負荷の状況になると、cnSGW-C がユーザープレーンに対する Sx Establishment Request メッセージのスロットリングを設定します。このスロットリングにより、過負荷のユーザープレーンへの新しいコール（低優先順位または非緊急）が回避されます。

スロットリングは、報告された OCI 値（パーセンテージ）に従って実行されます。スロットリングが発生すると、次のアクションが実行されます。

- そのユーザープレーンに対する報告された Sx Establishment Request メッセージのパーセンテージによるランダムなドロップ。
- `sx_no_resource_available` 切断理由での、cnSGW-C でのコールのドロップ。
- 各統計値の増加。

自己保護モードでのユーザープレーンからのセッション終了トリガー

ユーザープレーンは、Sx レポート要求メッセージを介し、ペーシング方法を使用して cnSGW-C に対するセッション終了要求をトリガーします。ユーザープレーンは、自己保護モードになっていて、負荷が改善されていない場合に、セッション終了要求をトリガーします。このトリガーは、SPTER（自己保護終了要求）ビットが設定されている場合に発生します。

cnSGW-C はそれらの PDN の Sx 終了要求を開始し、userplane_requested_termination という切断理由で PDN セッションを解放します。

輻輳原因に対する障害処理プロファイル サポート

ユーザーペインが自己保護モードで、PFCP_ENTITY_IN_CONGESTION (74) を原因として新しいセッションを拒否する場合、cnSGW-C は失敗テンプレートプロファイル構成に従って別のユーザープレーンを選択します。

障害対応プロファイルは UPF グループに関連付けられます。

UPF の再選択は UPF 選択プロセスに従って行われ、プロファイル構成とは異なる UPF への再試行回数が考慮されます。

（注） 現在、この機能の一部としてユーザープレーンを再試行および再選択するための原因コードでサポートされているのは、PFCP_ENTITY_IN_CONGESTION(74) のみです。

Sx 負荷/過負荷機能の設定

ここでは、Sx 負荷/過負荷の設定方法について説明します。

Sx 負荷/過負荷の設定を設定するには、次のコマンドを使用します。

```
config
 instance instance-id instance_id
   endpoint endpoint_name
    interface interface_name
     supported-features [ load-control | overload-control ]
    exit
   exit
```

注：

- **endpoint** *endpoint_name*：エンドポイント名を指定します。
- **interface** *interface_name*：インターフェイス名を指定します。
- **supported-features [load-control | overload-control]**：負荷/過負荷制御を有効にします。

設定例

次に設定例を示します。

```
configure
 instance instance-id 1
  endpoint pfcp
   interface sxa
    supported-features load-control overload-control
   exit
```

Sx 負荷/過負荷の設定の確認

次の `show` コマンドを使用して、Sx 負荷/過負荷の設定を表示します。

```
show running-config instance instance-id 1 endpoint
instance instance-id 1
endpoint pfcp
interface sxa
supported-features load-control overload-control
exit
exit
```

障害処理プロファイルの構成

ここでは、障害処理プロファイルの構成方法について説明します。

次のコマンドを使用して、障害処理プロファイルを構成します。

```
config
 profile failure-handling failure-handling_profile_name
   interface interface_name
    message message_type
      cause-code cause_code
        action action_type
          max-retry max_retry_count
      exit
    exit
   exit
 profile upf-group upf-group_profile_name
  failure-profile profile_name
 exit
```

注：

- **profile failure-handling** *failure-handling_profile_name*：障害処理プロファイル名を指定します。
- **interface** *interface_name*：インターフェイス名を指定します。
- **message** *message_type*：メッセージタイプを指定します。
- **cause-code** *cause_code*：原因 ID（2 ～ 255 の範囲）または原因 ID の範囲（2 ～ 255 の範囲）を「-」、「,」、またはその両方で区切って指定します。

  -または-

  以下のいずれかを指定する必要があります。

  - no-resource-available（リソースなし）
  - no-response-received（応答を受信しませんでした）
  - pfcp-entity-in-congestion（pfcp エンティティの輻輳）
  - reject
  - service-not-supported（サービスがサポートされていない）

- system-failure（システム障害）

- **action** *action_type*：原因のアクションタイプを指定します。以下のいずれかを指定する必要があります。
  - retry-terminate（再試行-終了）
  - terminate

- **max-retry** *max_retry_count*：retry-terminate アクションの最大再試行回数を指定します。0 ～ 5 の範囲の整数で指定する必要があります。デフォルト値は 1 です。
- **profile upf-group** *upf-group_profile_name*：UPF グループプロファイル名を指定します。
- **failure-profile** *profile_name*：UPF 障害プロファイル名を指定します。

設定例

次に構成例を示します。

```
profile failure-handling fh1
 interface sxa
  message SessionEstablishmentReq
   cause-code pfcp-entity-in-congestion action terminate
  exit
 exit
exit
profile failure-handling fh2
 interface sxa
  message SessionEstablishmentReq
   cause-code 74 action retry-terminate max-retry 3
  exit
 exit
exit
profile upf-group g1
 failure-profile fh1
exit
profile upf-group g2
 failure-profile fh2
exit
```

自己過負荷状態でのSx要求メッセージのスロットリングと除外

表 *238 :* 機能の履歴

機能名	リリース情報	説明
自己過負荷状態での着信 Sx メッセージの処理	2024.04.1	cnSGWc の負荷および過負荷制御機能が強化されて、自己過負荷状態時に着信 Sx メッセージを処理できるようになりました。 この機能により、cnSGWc は着信 DLDR Session Report Request メッセージを自己保護モードでスロットリングできます。 **デフォルト設定**：[Disabled]：有効にするには設定が必要。

着信 Sx 要求メッセージのスロットリングと除外

cnSGWc は、UPF がアイドルセッションのダウンリンクを受信するたびに、UPF から DDLR Session Report Request メッセージを受信します。cnSGWc は、自己保護モードになっていて、過負荷除外プロファイルが設定されている場合、DNN、ARP、QCI、およびメッセージの優先順位値について着信 DDLR Session Report メッセージを検証します。

Sxa インターフェイスの過負荷除外プロファイルを設定するには、「過負荷除外プロファイルの設定」を参照してください。

一致した場合、またはセッションが WPS または緊急セッションとして識別された場合、cnSGWc は着信 DLDR Session Report Request メッセージを処理します。それ以外の場合、**理由 74（PFCP Entity in Congestion**）でメッセージを拒否し、内部 DDN 自己保護タイマーを開始します。

内部 DDN 自己保護タイマーは、次のように機能します。

- 内部 DDN 自己保護タイマーが期限切れになる前に、cnSGWc の自己保護状態が解除されると、cnSGW は Downlink Data Notification メッセージを MME に送信します。
- 内部 DDN 自己保護タイマーが期限切れになり、cnSGWc が自己保護状態のままの場合、cnSGW は、DROBU フラグとダウンリンク FAR 適用アクションを BUFFER として、UPF に対して Sx Modification Request を送信します。
- DDN 自己保護タイマーが期限切れになる前に UE がアイドル状態を解除（MBReq を受信）すると、cnSGW は DDN 自己保護タイマーを停止して MBReq の処理を開始します。

（注） 自己保護モードの cnSGWc は、通常どおり、緊急、WPS セッションの DLDR Session Report Request、および DLDR 以外の Session Report Request を処理します。

発信 Sx 要求メッセージのスロットリングと除外

自己過負荷状態では、着信 GTPC インターフェイスにスロットリングが適用されるため、cnSGWc は、間接的に Sxa インターフェイス上の発信要求メッセージをスロットリングします。cnSGWc は、GTPC インターフェイス、および Sxa インターフェイスを介したスロットリングから除外されたメッセージを処理します。

Sx 負荷/過負荷制御 OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

UE Disconnect 統計

```
sgw_ue_disconnect_stats6app_name="smf",cluster="cn",data_center="cn",instance;_id=
 "0",reason="sx_no_resource_available",service_name="sgw-service"] 1
```

```
sgw_ue_disconnect_stats6app_name="smf",cluster="cn",data_center="cn",instance_id=
 "0",reason="userplane_requested_termination",service_name="sgw-service"] 1;
```

PDN Disconnect 統計

```
sgw_pdn_disconnect_stats6app_name="smf",cluster="cn",data_center="cn",instance_id=

"0",pdn_type="ipv4",rat_type="EUTRAN",reason="sx_no_resource_available",service_name="sgw-service
 " 1;
```

```
sgw_pdn_disconnect_stats6app_name="smf",cluster="cn",data_center="cn",instance_id=

"0",pdn_type="ipv4v6",rat_type="EUTRAN",reason="userplane_requested_termination",service_name="sgw-service
 " 1;
```

SGW サービス統計

```
sgw_service_stats6app_name="smf",cluster="cn",data_center="cn",fail_reason="sx_oci_throttling_reject",

instance_id="0",interface="interface_sgw_ingress",reject_cause="no_resources_available",service_name=
 "sgw-service ",sgw_procedure_type = "initial_attach",status =
"rejected",sub_fail_reason="" 1
```

```
sgw_service_stats"smf",cluster="cn",data_center="cn",fail_reason="0",instance_id="0",

interface="interface_sgw_egress",reject_cause="",service_name="sgwservice",sgw_procedure_type=
 "upf_initiated_deletion",status="attempted",sub_fail_reason="" 1;
```

```
sgw_service_stats6fail_reason="sx_cause_fail",interface="interface_sgw_ingress",reject_cause=

"service_denied",sub_fail_reason="pfcp_entity_in_congestion",sgw_procedure_type="initial_attach",
 status="rejected";
```

Session Report Request のスロットリング統計

新しいステータス文字列 **validation_failurePFCP_Entity_In_Congestion** が既存の統計 **sgw_sx_session_report_stats** に追加されました。cnSGWc は、DLDR Session Report Request が自己保護モードで拒否された場合に、このステータス値をセッションレポート統計に入力します。

```
sgw_sx_session_report_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",

instance_id="0",reason="",service_name="sgw-service",status="validation_failurePFCP_Entity_In_Congestion",
 sx_session_report_type="DLDR_NO_ARP"} 1
```

第 47 章

古いセッションの処理とクリア

- 機能の概要と変更履歴 （753 ページ）
- 機能説明 （754 ページ）
- 機能の仕組み （754 ページ）
- 機能設定 （757 ページ）
- OAM サポート （757 ページ）

機能の概要と変更履歴

要約データ

表 *239* : 要約データ

該当製品または機能エリア	cnSGW-C
該当するプラットフォーム	SMI
機能のデフォルト設定	無効：有効にするには設定が必要
関連資料	

マニュアルの変更履歴

表 *240* : マニュアルの変更履歴

改訂の詳細	リリース
初版	2021.02.3

機能説明

cnSGW-C は、SGW プロファイル構成の *session-stale-timer* パラメータを使用して、古いセッションの特定とクリアをサポートしています。古いセッションの例は、非アクティブで、特定の期間にわたって読み取りも変更もされていないセッションです。

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

タイマー期限切れ処理のコールフロー

ここでは、タイマー期限切れ処理のコールフローについて説明します。

図 *128* : タイマー期限切れ処理のコールフロー

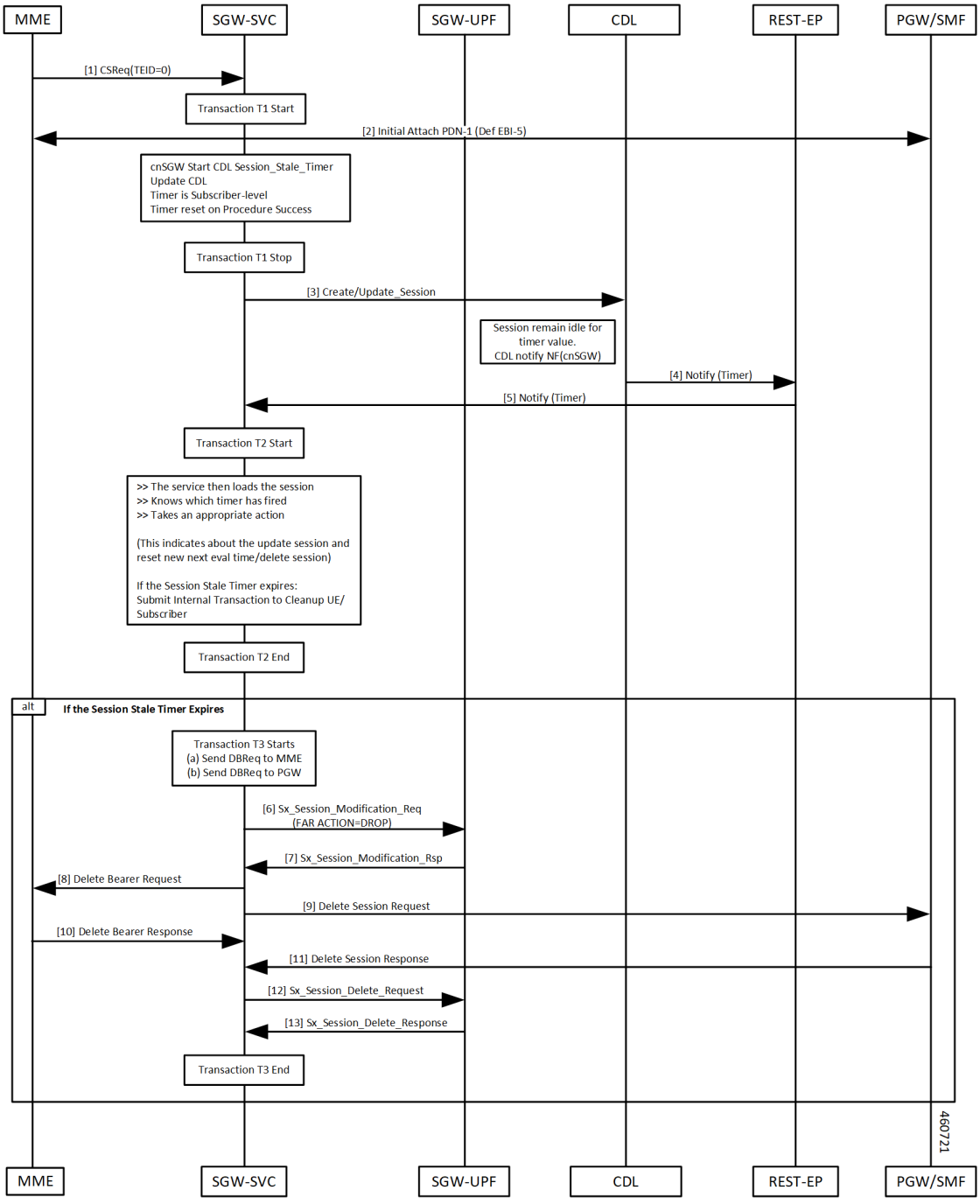


表 *241 :* タイマー期限切れ処理コールフローの説明

ステップ	説明
1、2、3	初期接続およびセッション失効タイマーの更新情報が CDL に送信され、タイマーが開始されます。手順は以下のとおりです。 • 初期接続が成功します。 • セッション失効タイマーが作成されます。 • CDL の更新が完了します。 • CDL は時間（評価時間）を開始し、更新セッションを待機します。 （注） コントロールプレーン（cnSGW-C）で任意のアクティビティまたはトランザクションのいずれかが発生した場合、セッション失効タイマーはリセットまたは再起動します。
4、5	CDL ポッドでタイマーが期限切れになり、cnSGW-C に通知を送信します。手順は次のとおりです。 • タイマー評価期間にセッション更新情報を受信しなかった場合。 • CDL ポッドのタイマーが期限切れになります。 • CDL が cnSGW-C にタイマー通知を送信します。
6 ～ 13	セッション失効タイマーが期限切れになり、タイマーのリセットが不要な場合、セッションクリーンアップがアクティブになります。手順は以下のとおりです。 • cnSGW-C でタイマー通知を受信します。 • タイマー通知がセッション失効タイマー用で、タイマーのリセットが不要な場合、UE セッションのクリーンアップ アクティビティが開始されます。 • Sx_Session_Modification_Req を UPF に送信して FAR Action=Drop を設定し、Sx_Session_Modification 応答を受信します。 • ベアラー削除要求を MME に送信します。 • セッション削除要求を PGW に送信します。 • Sx_Session_Delete_Request を UPF に送信して、ユーザープレーンデータをクリーンアップします。

機能設定

この機能を設定するには、次の設定を使用します。

```
config
   profile sgw sgw_profile_name
   session-stale-timer session_stale_timer
   end
```

注：

- **session-stale-timer** *session_stale_timer*：セッションがシグナリングまたはイベントなしでアイドル状態を維持できる最大秒数を指定します。この秒数を経過すると、セッションは終了します。

  session_stale_timer の値は、0 ～ 4294967295 の範囲で指定し、**setup-timeout** および **session-idle-timeout** タイマーの値より大きくする必要があります。

  session-stale-timer 設定を無効にするには、0 に設定します。

設定例

次に、設定例を示します。

```
config
   profile sgw sgw1
   session-stale-timer 120
   end
```

設定の確認

設定を確認するには、次のコマンドを実行します。

```
show running-config profile sgw sgw1
session-stale-timer 120
```

show コマンドの出力には、次のフィールドが含まれています。

session-stale-timer：セッションが終了した後、シグナリングまたはイベントなしでアイドル状態を維持できる最大秒数を示します。

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

この機能では、次の統計がサポートされています。

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",
fail_reason="",gr_instance_id="1",instance_id="0",interface="interface_sgw_egress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="stale_session_initiated_deletion",status="attempted",
 sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

gr_instance_id="1",instance_id="0",interface="interface_sgw_egress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="stale_session_initiated_deletion",status="success",
 sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

gr_instance_id="1",instance_id="0",interface="interface_sgw_ingress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="stale_session_initiated_deletion",status="attempted",
 sub_fail_reason=""} 1
```

```
sgw_service_stats{app_name="smf",cluster="Local",data_center="DC",fail_reason="",

gr_instance_id="1",instance_id="0",interface="interface_sgw_ingress",reject_cause="",

service_name="sgw-service",sgw_procedure_type="stale_session_initiated_deletion",status="success",
 sub_fail_reason=""} 1
```

```
sgw_ue_disconnect_stats{app_name="smf",cluster="Local",data_center="DC",
gr_instance_id="1",instance_id="0",reason="stale_session_init_disconnect",
service_name="sgw-service"} 1
```

```
sgw_pdn_disconnect_stats{app_name="smf",cluster="Local",data_center="DC",
gr_instance_id="1",instance_id="0",pdn_type="ipv4",rat_type="EUTRAN",
reason="stale_session_init_disconnect",service_name="sgw-service"} 1
```

第 48 章

CSFB 手順のサポート：中断と再開

- 機能の概要と変更履歴 （759 ページ）
- 機能説明 （760 ページ）
- 機能の仕組み （760 ページ）

機能の概要と変更履歴

要約データ

表 *242 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当するプラットフォーム	SMI
機能のデフォルト設定	[Enabled]：常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *243 :* マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2022.04.0

機能説明

回線交換フォールバック（CSFB）により、UE は EUTRAN セルに待機し、回線交換（CS）ドメインまたは他の CS ドメインサービス（ロケーションサービス（LCS）、補足サービスなど）への強制スイッチオーバーを介して音声コールを発信または終了できます。また、CS コアネットワークを介した SMS の配信は CSFB なしで実現されます。LTE EPC ネットワークは CS 接続を直接アンカーすることを意図していないため、CS 音声サービスが開始されると、EUTRAN ネットワーク上の PS ベースのデータアクティビティが一時停止します（つまり、データ転送が中断されるか、パケット交換接続が 2G/3G ネットワークに渡されます）。データアクティビティをさらに再開して、最初のサービス開始時に IMS パケット交換サービスを展開する予定のない LTE オペレータ向けにテレフォニーおよび SMS サービスを有効にすることができます。

cnSGW-C は、GTPC を介した S11 インターフェイスで次の CSFB メッセージングサービスをサポートします。

- 一時停止通知
- 一時停止確認応答
- 再開通知
- 再開確認応答

再開手順に加えて、MME が空でないベアラー変更要求を送信できる可能性があります。一時停止状態では、cnSGW-C は、これを暗黙的な再開手順と見なすことによって、ベアラー変更要求の処理をサポートします。

機能の仕組み

ここでは、この機能の仕組みを説明します。

cnSGW-C は、一時停止通知メッセージを PGW に転送し、非 GBR トラフィックのダウンリンクデータを一時停止します。その後、UE が CS サービスを完了して E-UTRAN に戻ると、MME は再開通知メッセージを cnSGW-C に送信し、cnSGW-C はそのメッセージを PGW に転送します。その後、ダウンリンク データ トラフィックが再開されます。

再開のための空のベアラー変更要求のサポート

一時停止状態では、cnSGW-C はベアラー変更要求を暗黙的な再開手順と見なして処理します。SGW は再開し、空のベアラー変更要求を PGW に転送します。eNodeB teid が MBR に存在する場合、cnSGW-C はダウンリンク方向の FAR アクション Forward に設定します。eNodeB teid が MBR に存在しない場合、cnSGW-C はダウンリンク方向の FAR アクションを Buffer に設定します。

コール フロー

ここでは、この機能の主要なコールフローについて説明します。

Suspend Notification コールフロー

ここでは、Suspend Notification コールフローについて説明します。

図 *129 : Change Notification Request* コールフロー

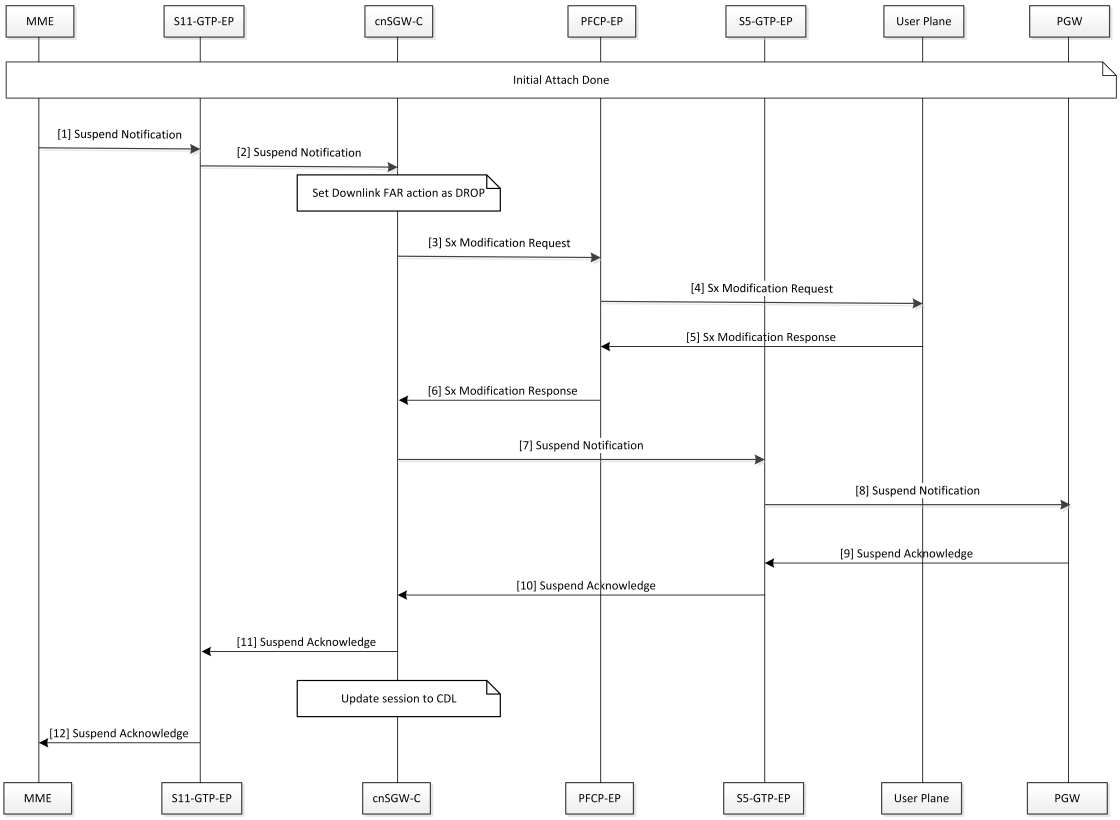


表 *244 : Suspend Notification* コールフローの説明

ステップ	説明
1	MME が Suspend Notification を S11-GTP-EP ポッドに送信します。
2	S11-GTP-EP ポッドが Suspend Notification を cnSGW-c に送信します。
3	cnSGW-c が、アクションセットを DROP にして Sx Session Modification Request を SGW-U に送信することで、ダウンロード FAR アクション Drop を更新します。 cnSGW-c が Sx Modification Request を PFCP-EP に送信します。

ステップ	説明
4	PFCP-EP が Sx Modification Request をユーザープレーンに送信します。
5	ユーザープレーンが Sx Modification Response を PFCP-EP に送信します。
6	PFCP-EP が Sx Modification Response を cnSGW-C に送信します。
7	cnSGW-C が Suspend Notification を S5-GTP-EP ポッドに送信します。
8	S5-GTP-EP ポッドが Suspend Notification を PGW に送信します。
9	PGW が Suspend Acknowledgment を S5-GTP-EP ポッドに送信します。
10	S5-GTP-EP ポッドが Suspend Acknowledgment を cnSGW-C に送信します。
11	cnSGW-C が Suspend Acknowledgment を S11-GTP-EP ポッドに送信します。 cnSGW-C が CDL へのセッションを更新します。
12	S11-GTP-EP ポッドが Suspend Acknowledgment を MME に送信します。

再開通知のコールフロー

このセクションでは、再開通知のコールフローについて説明します。

図 *130 :* 再開通知のコールフロー

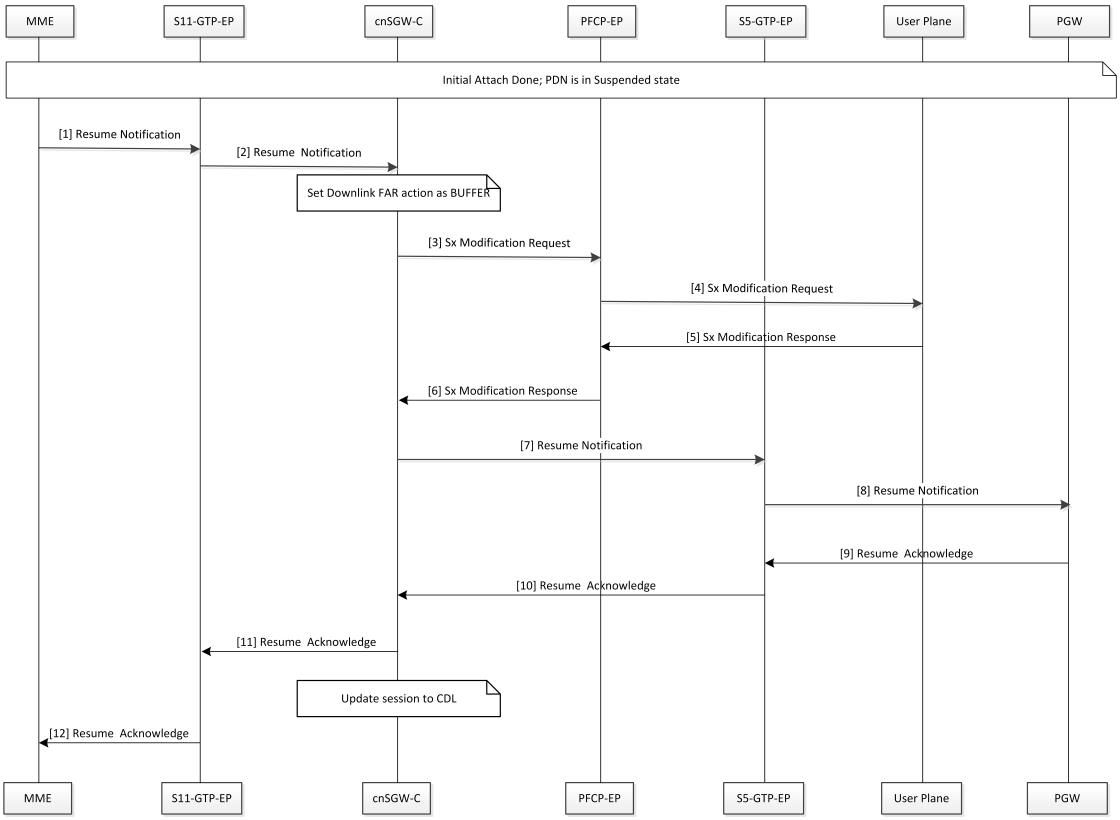


表 *245 :* 再開通知のコールフローの説明

ステップ	説明
1	MME が再開通知を S11-GTP-EP ポッドに送信します。
2	S11-GTP-EP ポッドが再開通知を cnSGW-c に送信します。
3	cnSGW-c が、アクションセットを BUFFER として Sx Session Modification Request を SGW-Uに送信することで、FARのダウンロードアクションのバッファを更新します。 cnSGW-c が Sx 変更要求を PFCP-EP に送信します。
4	PFCP-EP が Sx 変更要求をユーザープレーンに送信します。
5	ユーザープレーンが PFCP-EP に Sx 変更応答を送信します。
6	PFCP-EP が cnSGW-C に Sx 変更応答を送信します。
7	cnSGW-C が再開通知を S5-GTP-EP ポッドに送信します。

ステップ	説明
8	S5-GTP-EP ポッドが再開通知を PGW に送信します。
9	PGW が再開確認応答を S5-GTP-EP ポッドに送信します。
10	S5-GTP-EP ポッドが再開確認応答を cnSGW-C に送信します。
11	cnSGW-C が再開確認応答を S11-GTP-EP ポッドに送信します。 cnSGW-C が CDL へのセッションを更新します。
12	S11-GTP-EP ポッドが再開確認応答を MME に送信します。

再開用の空のベアラー変更要求のコールフロー

このセクションでは、再開用の空のベアラー変更要求について説明します。

図 *131* : 再開用の空のベアラー変更要求のコールフロー

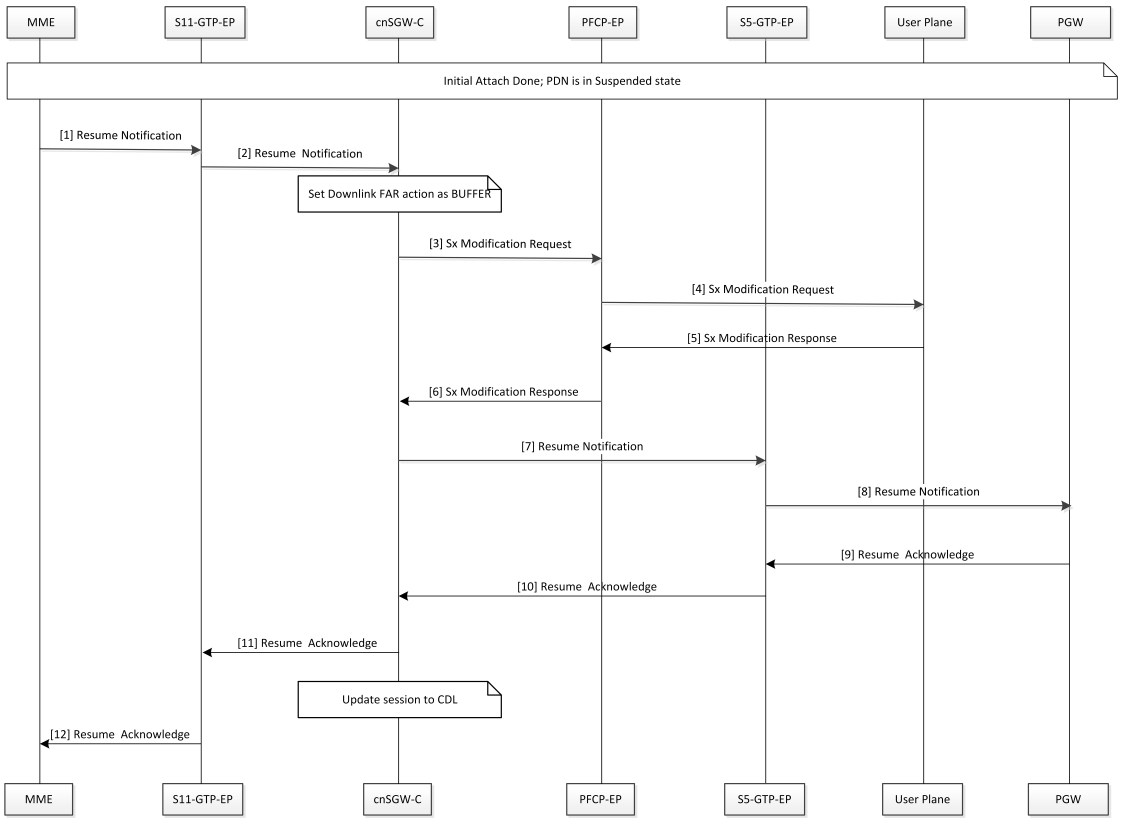


表 *246 :* 再開用の空のベアラー変更要求のコールフローの説明

ステップ	説明
1	MME が S11-GTP-EP ポッドにベアラー変更要求を送信します。
2	S11-GTP-EP ポッドが cnSGW-C にベアラー変更要求を送信します。 cnSGW-C はこの要求を SGW で一時停止されたベアラーの暗黙的な再開と見なします。
3	eNodeB が存在する場合、cnSGW-C はダウンリンク方向の FAR アクションの設定を Forward に更新します。それ以外の場合、cnSGW-C はダウンリンク方向の FAR アクションの設定を BUFFER に更新します。 cnSGW-C が PFCP-EP ポッドに Sx 変更要求を送信します。
4	PFCP-EP がユーザープレーンに Sx 変更要求を送信します。
5	ユーザープレーンが PFCP-EP に Sx 変更応答を送信します。
6	PFCP-EP が cnSGW-C に Sx 変更応答を送信します。
7	cnSGW-C が S5-GTP-EP ポッドにベアラー変更要求を送信します。 PGW の暗黙的な再開には、0 個以上の IE を含めることができます。
8	S5-GTP-EP ポッドが PGW にベアラー変更要求を送信します。
9	PGW が S5-GTP-EP ポッドにベアラー変更応答を送信します。
10	S5-GTP-EP ポッドが cnSGW-C にベアラー変更要求を送信します。
11	cnSGW-C が S11-GTP-EP ポッドにベアラー変更応答を送信します。
12	S11-GTP-EP ポッドが MME にベアラー変更応答を送信します。

第 49 章

ベアラー更新要求および応答

- 機能の概要と変更履歴 （767 ページ）
- 機能説明 （768 ページ）
- 機能の仕組み （768 ページ）

機能の概要と変更履歴

要約データ

表 *247 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *248 :* マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	2020.04

機能説明

このリリースでは、cnSGW-C は、PGW から MME への更新ベアラー要求（TFT の変更、QCI の変更、または APB - AMBR の変更を含むことができる）のリレーのみをサポートします。MME は cnSGW-C に応答を送信すると、更新ベアラー応答を PGW に向けてリレーします。

このリリースでは、ユーザープレーンへのシグナリングはサポートされていません。

標準準拠

Update Bearer Request および Response Support 機能は、次の標準に準拠しています。

- *3GPP TS 23.401*
- *3GPP TS 23.214*
- *3GPP TS 29.274*
- *3GPP TS 29.244*

機能の仕組み

ここでは、この機能の仕組みを説明します。

コール フロー

ここでは、Update Bearer Request および Response 機能の主要なコールフローについて説明します。

図 *132 : UP* シグナリングなしの *Bearer Request/Response* コールフロー

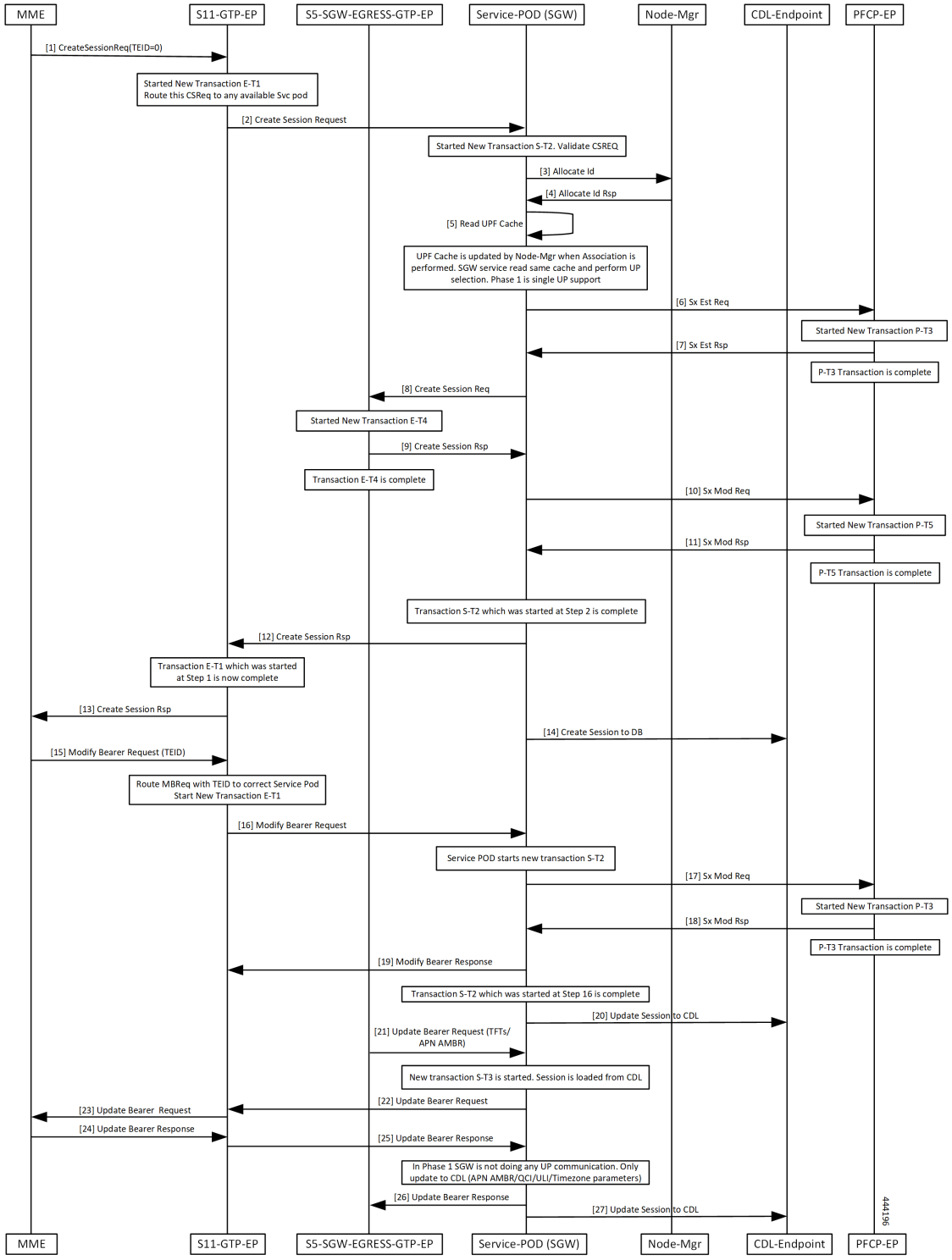
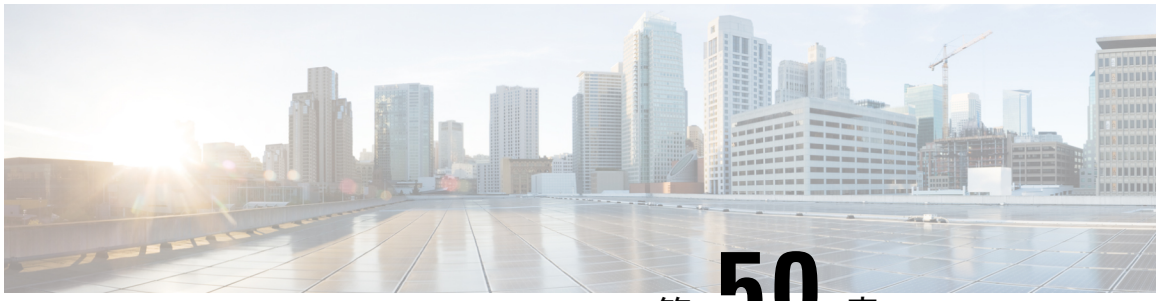


表 *249 : UP* シグナリングなしの *Bearer Request/Response* コールフローの説明

ステップ	説明
1	MME が TEID 値がゼロの Create Session Req を S11-GTP-EP に送信します。
2	トランザクション E-T1 が開始しました。 S11-GTP-EP が Create Session Req を SGW サービスポッドに送信します。
3、4	トランザクション S-T2 が開始しました。 SGW サービスポッドが割り当て ID を Node-Mgr に送信し、応答を受信します。
5、6、7	SGW サービスポッドが Node-Mgr で更新された UPF キャッシュを読み取り、UPF 選択を実行します。 SGW サービスポッドが Sx Est Req を PFCP-EP に送信します。 トランザクション P-T3 が開始しました。 SGW サービスポッドが PFCP-EP から Sx Est Rsp を受信します。
8、9	トランザクション P-T3 が完了しました。 SGW サービスポッドが Create Session Req を S5-SGW-EGRESS-GTP-EP に送信します。 トランザクション E-T4 が開始しました。 SGW サービスポッドが S5-SGW-EGRESS-GTP-EP から Create Session Rsp を受信します。
10、11、12	トランザクション E-T4 が完了しました。 SGW サービスポッドが Sx Mod Req を PFCP-EP に送信します。 トランザクション P-T5 が開始しました。 SGW サービスポッドが PFCP-EP から Sx Mod Rsp を受信します。
12	トランザクション S-T2 と P-T5 が完了しました。 SGW サービスポッドが Create Session Rsp を S11-GTP-EP に送信します。
13	トランザクション E-T1 が完了しました。 S11-GTP-EP が Create Session Rsp を MME に送信します。
14	SGW サービスポッドが Create Session を DB に送信します。
15	MME が TEID 値とともに Modify Bearer Req を S11-GTP-EP に送信します。

ステップ	説明
16	S11-GTP-EP が TEID とともに MBReq を正確な SGW サービスポッドにルーティングします。 S11-GTP-EP が Modify Bearer Request を SGW サービスポッドに送信します。
17	トランザクション S-T2 が完了しました。 SGW サービスポッドが Sx Mod Req を PFCP-EP に送信します。
18	トランザクション P-T3 が開始しました。 SGW サービスポッドが PFCP-EP から Sx Mod Rsp を受信します。
19	トランザクション P-T3 が完了しました。 SGW サービスポッドが Sx Mod Rsp を S11-GTP-EP に送信します。
20	トランザクション S-T2 が完了しました。 SGW サービスポッドが更新セッションを CDL に送信します。
21	S5-SGW-EGRESS-GTP-EP が TFT および APN AMBR とともに Update Bearer Request を SGW サービスポッドに送信します。
22、23	トランザクション S-T3 が完了しました。 SGW サービスポッドが Update Bearer Request を S11-GTP-EP に送信します。 S11-GTP-EP が Update Bearer Request を MME に転送します。
24、25	MME が Update Bearer Rsp を S11-GTP-EP に送信します。 S11-GTP-EP が Update Bearer Rsp を SGW サービスポッドに転送します。
26	SGW サービスポッドが Update Bearer Rsp を S5-SGW-EGRESS-GTP-EP に転送します。
27	SGW サービスポッドが更新セッションを CDL に送信します。



第 50 章

UPF 選択のサポート

- 機能の概要と変更履歴 （773 ページ）
- 機能説明 （774 ページ）
- DNN および DCNR サポートを使用した UPF の選択 （774 ページ）
- ロケーションサポートを使用した UPF の選択 （779 ページ）
- cnSGW-C と SMF の UPF の組み合わせ選択 （784 ページ）
- UPF 選択 OAM のサポート （795 ページ）

機能の概要と変更履歴

要約データ

表 *250 :* 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	DCNR サポートを使用した UPF 選択：無効：有効にするには設定が必要です DNN サポートを使用した UPF 選択：有効：常時オン ロケーションサポートを使用した UPF 選択：無効：有効にするには設定が必要です cnSGW-C と SMF の複合 UPF 選択：無効：有効にするには設定が必要です
関連資料	該当なし

マニュアルの変更履歴

表 *251 :* マニュアルの変更履歴

改訂の詳細	リリース
ロケーションを使用した UPF 選択のサポートが追加されました。 cnSGW-C および SMF の複合 UPF 選択のサポートが追加されました。	2021.02.0
最初の導入。	2021.01.0

機能説明

この機能では、次の UPF の選択方法について説明します。

- DNN と DCNR
- ロケーションサポート
- 同じ UPF インスタンスを選択する cnSGW-C と SMF

DNN および DCNR サポートを使用した UPF の選択

機能説明

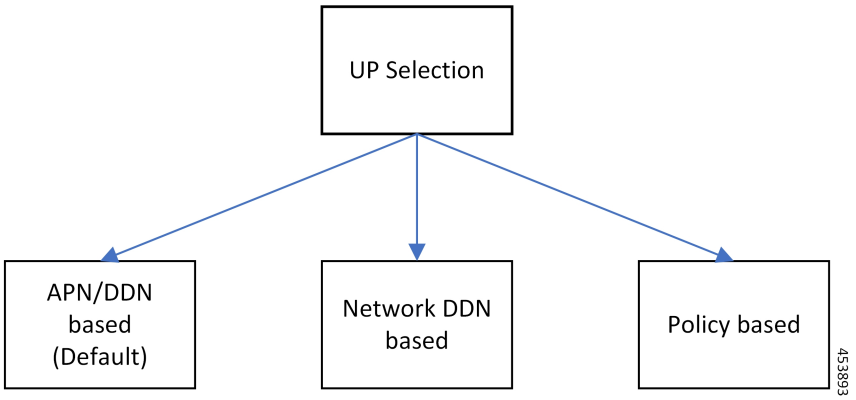
UPF の選択方法は次の 3 つです。

- DNN または APN ベース
- ネットワークベース
- ポリシーベース

（注） UPF 選択ポリシーが関連付けられていない場合、DNN は有効になります。

機能の仕組み

ここでは、この機能の仕組みを説明します。



UPF の選択方法

DNN または APN ベース

- Create Session Request メッセージには APN 情報が含まれています。この APN は、各ユーザープレーンのネットワーク要素プロファイルの DNN リストの一部として設定されます。
- PDN の確立では、それらのユーザープレーンが考慮されます。
- 多くのユーザープレーンが使用可能な場合、UPF 選択ではキャパシティと優先順位が使用されます。

ネットワークベース

- UPF の選択では、APN または DNN プロファイルの一部として設定された DNN が考慮されます。
- この DNN は、ローカル SGW ネットワーク固有の DNN 名です。
- 同じネットワーク DNN または APN 名が、各ユーザープレーンのネットワーク エレメント プロファイル内の DNN リストの一部として設定されます。
- CSReq に含まれる APN を使用する代わりに、DNN リストに基づいてローカルの DNN が UPF の選択に使用されます。

  たとえば、APN が不明なローミングシナリオの場合、この設定は UPF の選択に役立ちます。
- PDN の確立では、これらのユーザープレーンが考慮されます。
- 多くのユーザープレーンが使用可能な場合、UPF 選択ではキャパシティと優先順位が使用されます。

UPF 選択ポリシーベース

- パラメータを使用した UPF 選択プロファイルの設定により、各優先順位の UPF が決まります。優先順位でサポートされている最大数は 4 です。

- 各優先順位パラメータは、論理 AND 条件です。DNN と DCNR が優先順位 1 として設定されている場合、DNN でサポートされているユーザープレーンが検索され、DCNR ベースのサポートが有効になります。この検索条件が見つからない場合、次（ほとんどは 2）の優先順位に移動し、その条件の評価を試みます。
- UPF 選択ポリシーは、DNN プロファイルに関連付けられています。
- UPF グループは、同じ UPF グループプロファイルに属するネットワーク要素プロファイルに特性を提供します。
- 多くのユーザープレーンが利用可能な場合、UPF 選択ではキャパシティと優先順位を使用します。

（注）
- cnSGW-C は、優先順位基準と一致する UPF がない場合、原因を NO_RESOURCE_AVAILABLE に指定したセッション作成応答によりコールを拒否します。

UPF 選択方式の構成

ここでは、UPF 選択方法の構成方法について説明します。

UPF グループプロファイルベースの UPF 選択の設定

ここでは、UPF グループプロファイルベースの UPF 選択の設定方法について説明します。

UPF グループプロファイルベースの UPF 選択を設定するには、次のコマンドを使用します。

```
config
  profile upf-group upf_group_name
    dcnr [true | false]
    end
```

注：

- **profile upf-group** *upf_group_name*：UPF グループ名を指定します。文字列で指定する必要があります。
- **dcnr [true | false]**：新しい無線とのデュアル接続のサポートを有効または無効に指定します。デフォルト値は false です。

設定例

次に設定例を示します。

```
config
profile upf-group G1
 dcnr true
 end
```

ネットワークベースの UPF 選択の設定

表 *252 :* 機能の履歴

機能名	リリース情報	説明
データプレーンに対するデュアルスタックのサポート	2024.02.0	cnSGW-c は、UPF ネットワークプロファイルで **dual-stack-transport { false | true }** CLI コマンドを使用して、データプレーンのデュアル スタック トランスポートを有効にします。 このサポートにより、次のことが可能になります。 • ネットワーク拡張のために、IPv6 アドレスを使用して新しい eNB と UPF を設定する。 • IPv6 アドレスへの段階的な移行のために、IPv4 アドレスを使用して既存の eNB と UPF を続行する。 **デフォルト設定**：[Disabled]：設定が必要

次のコマンドを使用して、ネットワークベースの UPF 選択を設定します。

```
config
   profile network-element upf upf_name
      node-id  node_id_value
      n4-peer-address ipv4  ipv4_address
      n4-peer-address ipv6  ipv6_address
      n4-peer-port  port_number
      dual-stack-transport { true | false }
      upf-group-profile upf_group_profile_name
      dnn-list dnn_list
      priority priority_value
      capacity capacity_value
      end
```

注：

- **network-element upf** *upf_name*：UPF プロファイル名を指定します。
- **node-id** *node_id_value*：UPF ノードのノード ID を指定します。

- **n4-peer-address ipv4** *ipv4_address*：IPv4 アドレスを指定します。
- **n4-peer-address ipv6** *ipv6_address*：IPv6 アドレスを指定します。
- **dual-stack-transport { true | false }**：IPv6 または IPv4 アドレスを指定できるデュアルスタック機能を有効にします。この機能を有効にするには、true を指定します。
- **upf-group-profile** *upf_group_profile_name*：UPF グループプロファイル名を指定します。
- **dnn-list** *dnn_list*：UPF ノードでサポートされている DNN リストを指定します。
- **priority** *priority_value*：他の UPF に対する静的優先順位を指定します。この値はロードバランシングに使用され、0 ー 65535 の範囲の整数で指定する必要があります。デフォルト値は 1 です。
- **capacity** *capacity_value*：他の UPF に対するキャパシティを指定します。この値はロードバランシングに使用され、0 ー 65535 の範囲の整数で指定する必要があります。デフォルト値は 10 です。

設定例

以下に設定例を示します。

```
config
profile network-element upf UP1
 node-id      upf1@sgw.com
 upf-group-profile G1
 dnn-list [dnn1 dnn2]
 priority 20
 capacity 65535
 end
```

ポリシーベースの UPF 選択の設定

ここでは、ポリシーベースの UPF 選択の設定方法について説明します。

ポリシーベースの UPF 選択を設定するには、次のコマンドを使用します。

```
config
 policy upf-selection upf_selection_policyname
   precedence precedence_value location
   exit
   precedence precedence_value dnn
   exit
 exit
```

注：

- **upf-selection** *upf_selection_policyname*：UPF 選択ポリシー名を指定します。
- **precedence** *precedence_value*：エントリの優先順位を指定します。1 ～ 4 の範囲の整数である必要があります。

設定例

次に設定例を示します。

```
config
policy upf-selection upf_pol1
  precedence 1
    [ location ]
  exit
  precedence 2
    [ dnn ]
  exit
exit
```

トラブルシューティング情報

ここでは、DNN および DCNR の設定の問題を使用して UPF 選択を表示できるようにするトラブルシューティング情報について説明します。

設定エラー

```
show config-error | tab
ERROR COMPONENT   ERROR DESCRIPTION
-------------------------------------------------------------------------------------------------------
SGWProfile        Subscriber policy name : sub_policy in profile sgw1 is not configured
SubscriberPolicy  Operator policy : op_policy1 under subscriber policy sub_policy2 is
not configured
OperatorPolicy    Dnn policy name : dnn_policy1 in operator policy op_policy2 is not
configured
DnnPolicy         Dnn profile name : dnn_profile1 in dnn policy dnn_policy2 is not
configured
DnnProfile       UPF selection policy name : upf_sel_policy1 in dnn profile dnn_profile2
 is not configured
```

ロケーションサポートを使用した UPF の選択

機能説明

この機能は、セッション作成要求メッセージでロケーションベースの UPF の選択をサポートしています。この選択は、受信した TAI または ECGI あるいは TAI と ECGI の両方の値に従って実行されます。

UPF 選択機能の設定

ここでは、ロケーションを使用した UPF 選択の設定方法について説明します。

EPS 用の ECGI の構成

ここでは、EPS の ECGI の構成方法について説明します。

個々の ECGI 値または ECGI の範囲のリストを構成するために、新しい構成とプロファイル **ecgi-group** が追加されました。

ECGI リストと ECGI 範囲の両方を構成できます。ECGI 範囲の構成はオプションです。

次のコマンドを使用して、EPS の ECGI 構成を構成します。

```
config
  profile ecgi-group ecgi_group_name
    mcc mcc_value
    mnc mnc_value
      ecgi list ecgi_list_name
      ecgi range start start_value end end_value
  exit
```

注：

- **ecgi-group** *ecgi_group_name*：ECGI グループ名を指定します。
- **mcc** *mcc_value*：MCC 値を指定します。3 桁の数字で指定する必要があります。例：123
- **mnc** *mnc_value*：MNC 値を指定します。2 桁または 3 桁の数字で指定する必要があります。例：23 または 456
- **ecgi list** *ecgi_list*：ECGI 値のリストを指定します。7 桁の 16 進文字列の EUT セル ID。たとえば、A12345f です。文字列で指定する必要があります。
- **ecgi range start** *start_valueend_value* **end**： ECGI 範囲の開始値と終了値を指定します。文字列で指定する必要があります。

（注）

- 複数の ECGI 範囲値を構成できます。
- **ecgi-group** 構成で複数の [PLMN and ECGI values] を構成できます。
- **ecgi-group** 設定では、最大 16 の PLMN を設定できます。
- PLMN の ECGI リストでは最大 64 の ECGI 値を構成できます。
- PLMN で定義される ECGI 範囲の最大定義数は 64 です。

設定例

次に設定例を示します。

```
config
profile ecgi-group e1 mcc 123 mnc 45
ecgi list [ 1234567 abcdef0 ]
ecgi range start 1111111 end fffffff
exit
```

EPS 用 ECGI の構成の確認

このセクションでは、EPS の ECGI 構成を確認する方法について説明します。

次の show コマンドを使用して、EPS の ECGI 構成を表示します。

```
show running-config profile ecgi-group
profile ecgi-group e1
mcc 123 mnc 45
ecgi list [ 1234567 abcdef0 ]
ecgi range start 1111111 end fffffff
exit
exit
exit
```

TAI グループの設定

ここでは、TAI グループの設定方法について説明します。

次の TAI グループ設定を拡張して、名前が異なる複数の TAI グループ設定をサポートできます。

次のコマンドを使用して、TAI グループを設定します。

```
config
  profile tai-group tai_group_name
    mcc mcc_value
    mnc mcc_value
      tac list tac_list
      tac range start start_value end end_value
  exit
```

注：

- **tai-group** *tail_group_name*：TAI グループ名を指定します。
- **mcc** *mcc_value*：MCC 値を指定します。3 桁の数字で指定する必要があります。例：123
- **mnc** *mnc_value*：MNC 値を指定します。2 桁または 3 桁の数字で指定する必要があります。例：23 または 456
- **tac list** *tac_list*：TAC 値のリストを指定します（[0-9a-fA-F]{4}|[0-9a-fA-F]{6}）。4 桁または 6 桁の 16 進文字列（例：A123、1a2B3F）。文字列で指定する必要があります。
- **tac range start** *start_value* **end** *end_value*：TAC 範囲の開始値と終了値を指定します。文字列で指定する必要があります。

（注）
- TAI グループの下に最大 16 の PLMN を設定できます。
- PLMN の下の TAC リストに最大 64 の TAC 値を設定できます。
- PLMN で定義される TAC 範囲の最大数は 64 です。

設定例

次に設定例を示します。

```
config
profile tai-group TAI-GRP1
  mcc 123 mnc 234
    tac list [ 1a25 A123 ]
    tac range start B234 end b999
    exit
    tac range start C213 end c999
    exit
  exit
  mcc 231 mnc 45
    tac list [ 2a2B B123 ]
    tac range start d111 end d999
    exit
  exit
exit
```

Location-area-group の設定

ここでは、Location-area-group の設定方法について説明します。

新しい設定とプロファイルの location-area-group を追加できます。**ecgi-group** と **tail-group** の設定は任意です。

次のコマンドを使用して、Location-area-group を設定します。

```
config
   profile location-area-group location_area_group_name
      tai-group tai_group_name
      ecgi-group ecgi_group_name
   exit
```

注：

- **location-area-group** *location_area_group_name*：ロケーションエリアグループ名を指定します。
- **tai-group** *tail_group_name*：TAI グループ名を指定します。
- **ecgi-group** *ecgi_group_name*：ECGI グループ名を指定します。

設定例

次に設定例を示します。

```
config
profile location-area-group LOC_AREA_GRP_1
  tai-group TAI-AUTO-GRP1
  ecgi-group ECGI-AUTO-GRP1
exit
profile location-area-group LOC_AREA_GRP_2
  tai-group TAI-AUTO-GRP2
exit
```

UPF グループと UPF 選択ポリシーの機能拡張の構成

ここでは、UPF グループと UPF 選択ポリシーの拡張機能を構成する方法について説明します。

upf-group-profile の下に新しい構成を追加して、location-area-group-list を構成できます。

次のコマンドを使用して、UPF グループおよび UPF 選択ポリシーの拡張を構成します。

```
config
  profile upf-group upf_group_name
     location-area-group-list [area_group_list]
  exit

config
  policy upf-selection selection_policy_name
   precedence value  [ selection_parameter_list ]
  exit
```

注：

- **upf-group** *upf_group_name*：UPF グループ名を指定します。
- **location-area-group-list** *area_group_list*：UPF ノードがサポートするロケーションエリアグループのリストを指定します。
- **upf-selection** *selection_policy_name*：UPF 選択ポリシー名を指定します。
- **precedence** *value* **[** *selection_parameter_list* **]**：エントリの優先順位を指定します。1 ～ 4 の範囲の整数である必要があります。

（注） pdn-type-subscription と pdn-type-session の両方が構成されている場合、pdn-type-subscription が考慮されます。

設定例

次に設定例を示します。

```
config
profile upf-group G1
 location-area-group-list [ LOC_AUTO_GRP_1 ]
exit
profile upf-group G2
 location-area-group-list [ LOC_AUTO_GRP_2 ]
exit
profile upf-group G3
 location-area-group-list [LOC_AUTO_GRP_1 LOC_AUTO_GRP_2 ]
exit

config
policy upf-selection upf_pol1
  precedence 1
    [ location ]
  exit
  precedence 2
    [ dnn ]
```

```
  exit
exit
```

cnSGW-C と SMF の UPF の組み合わせ選択

機能説明

この機能は、cnSGW-C と SMF をサポートしており、UPF と SMF が同じクラスタに展開され、UPF インスタンスが利用可能な場合に、同じ UPF インスタンスを選択できます。UPF インスタンスが使用できない場合、UPF は既存の構成に基づいて選択されます。

標準準拠

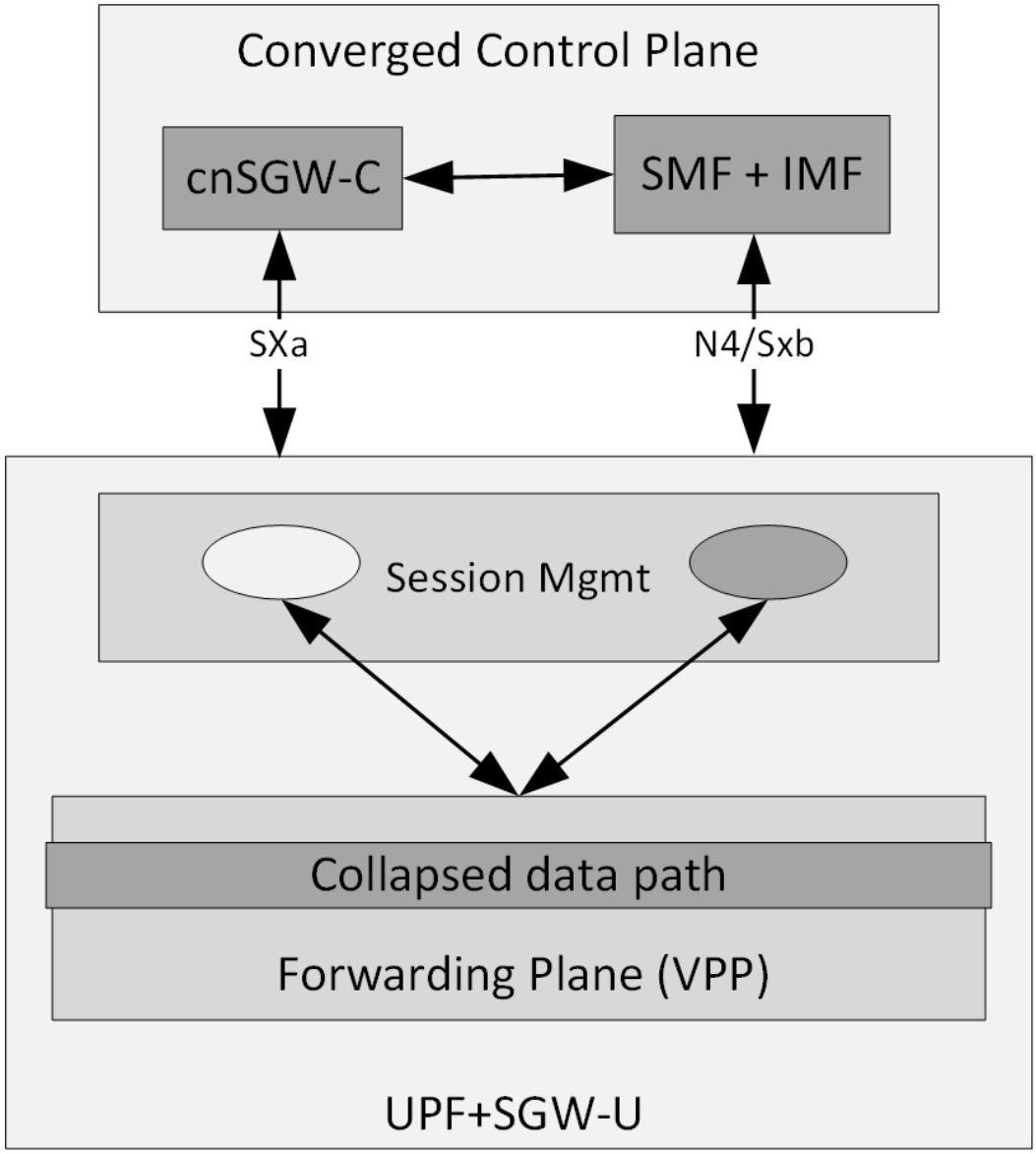
cnSGW-C と SMF の複合 UPF 選択機能は、次の標準に準拠しています。

- *3GPP TS 23.401*
- *3GPP TS 23.402*
- *3GPP TS 29.274*
- *3GPP TS 23.214*
- *3GPP TS 29.244*
- *3GPP TS 24.008*

機能の仕組み

ここでは、この機能の仕組みを説明します。

システム アーキテクチャ



cnSGW-C と SMF/IWF は同じ UPF インスタンスを使用するため、UPF はこれらのセッションをコラプストデータパスに使用できます。

コントロールプレーン（cnSGW-C および SMF）は、さまざまなシナリオ（初期接続、ハンドオーバーなど）で同じユーザープレーンを選択します。

初期接続中に次のアクションが実行されます。

- cnSGW-C は、初期接続時に、選択した UPF インスタンスの SGW-U FQDN 情報を SMF に渡します。
- SMF は、受信した SGW-U FQDN に従って UPF インスタンスを選択します。

- 同じ UPF FQDN が cnSGW-C と SMF で設定され、ネットワーク要素プロファイルの一部として関連付けが作成されます。

コール フロー

ここでは、cnSGW-C および SMF 機能の複合 UPF 選択の主要なコールフローについて説明します。

5G 対応デバイスのコールフローの 4G での初期接続

このセクションでは、5G 対応デバイスの 4G での初期接続のコールフローについて説明します。

図 *133 : 5G* 対応デバイスのコールフローの *4G* での初期接続

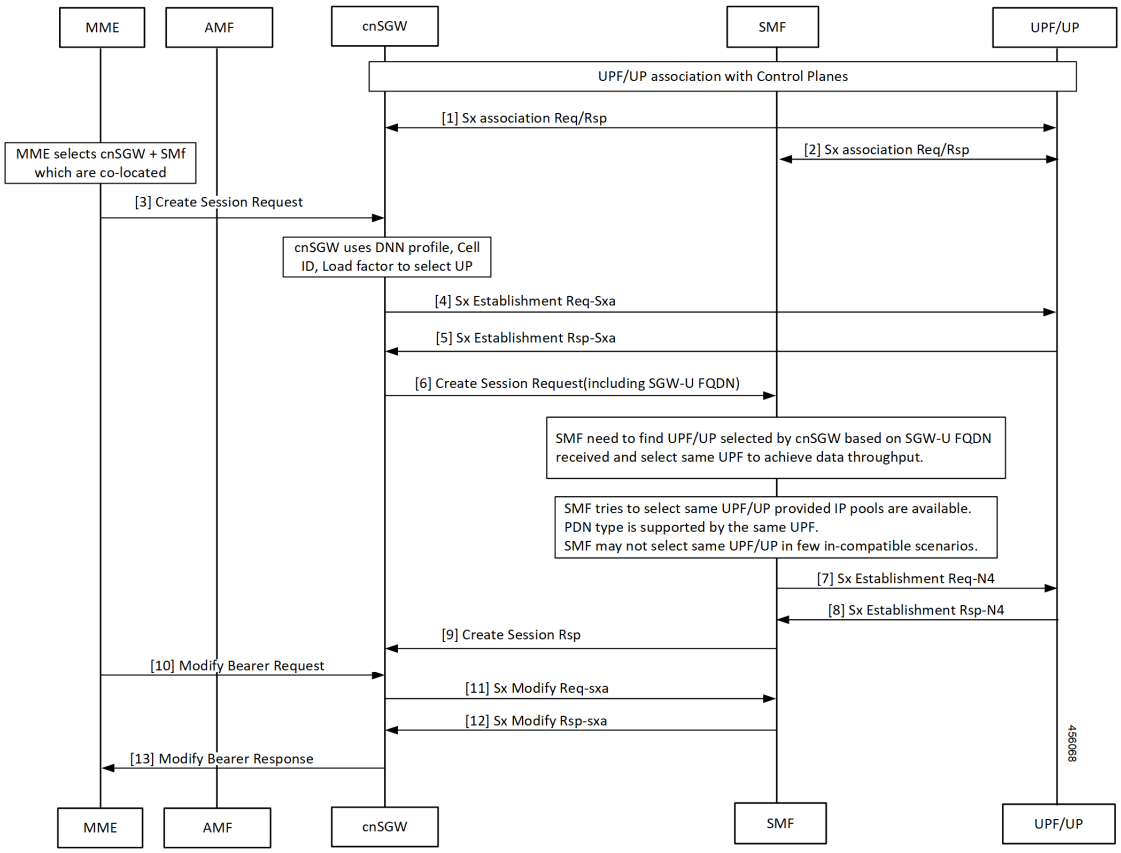


表 *253 : 5G* 対応デバイスの *4G* での初期接続コールフローの説明

ステップ	説明
1	コントロールプレーンとの UPF アソシエーションが確立されます。 cnSGW-C が Sx アソシエーション要求/応答を UPF に送信します。

ステップ	説明
2	SMF が Sx アソシエーション要求/応答を UPF に送信します。
3	MME が、同じ場所にある cnSGW-C と SMF を選択した後、cnSGW-C にセッション作成要求を送信します。
4	cnSGW-C が、DNN プロファイル、Cell ID、およびローカル要因を使用して UPF を選択した後、Sx 確立要求（SXA）を UPF に送信します。
5	cnSGW-C が UPF から Sx 確立応答を受信します。
6	cnSGW-C が、SGW-U FQDN を含むセッション作成要求を SMF に送信します。
7	SMF が、受信した SGW-U FQDN に従って cnSGW-C が選択した UPF を見つけ、同じ UPF を選択してデータスループットを実現する必要があります。 IP プールが使用可能な場合、SMF は同じ UPF を選択しようとします。同じ UPF で PDN タイプがサポートされます。 一部の互換性のないシナリオでは、SMF は同じ UPF を選択しない場合があります。 SMF が Sx 確立要求 N4 を UPF に送信します。
8	SMF が UP から Sx 確立応答 N4 を受信します。
9	cnSGW-C が、SMF からセッション作成応答を受信します。
10	MME が、ベアラー変更要求を cnSGW-C に送信します。
11	cnSGW-C が Sx 変更要求（SXA）を SMF に送信します。
12	cnSGW-C が、SMF への Sx 変更応答（SXA）を受信します。
13	cnSGW-C が、ベアラー変更応答を MME に送信します。

ユーザープレーン ID を使用した UPF 登録コールフロー

ここでは、ユーザープレーン ID を使用した UPF 登録コールフローについて説明します。

図 *134 :* ユーザープレーン *ID* を使用した *UPF* 登録コールフロー

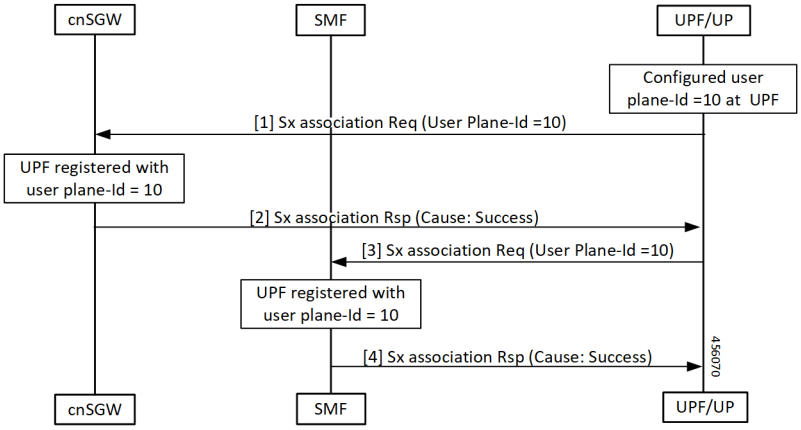


表 *254 :* ユーザープレーン *ID* を使用した *UPF* 登録コールフローの説明

ステップ	説明
1	UPF で設定されたユーザープレーン ID cnSGW-C が、設定されたユーザープレーン ID を持つ UPF から Sx 関連付け要求を受信します。
2	UPF が、cnSGW-C から原因 = SUCCESS の Sx 関連付け応答を受信します。
3	SMF が、設定されたユーザープレーン ID を持つ UPF から Sx 関連付け要求を受信します。
4	UPF が、SMF から原因 = SUCCESS の Sx 関連付け応答を受信します。

5G 対応デバイスの 4G RAT での SGW 間ハンドオーバーコールフロー

ここでは、5G 対応デバイスの 4G RAT での SGW 間ハンドオーバーコールフローについて説明します。

図 *135 : 5G* 対応デバイスの *4G RAT* での *SGW* 間ハンドオーバーコールフロー

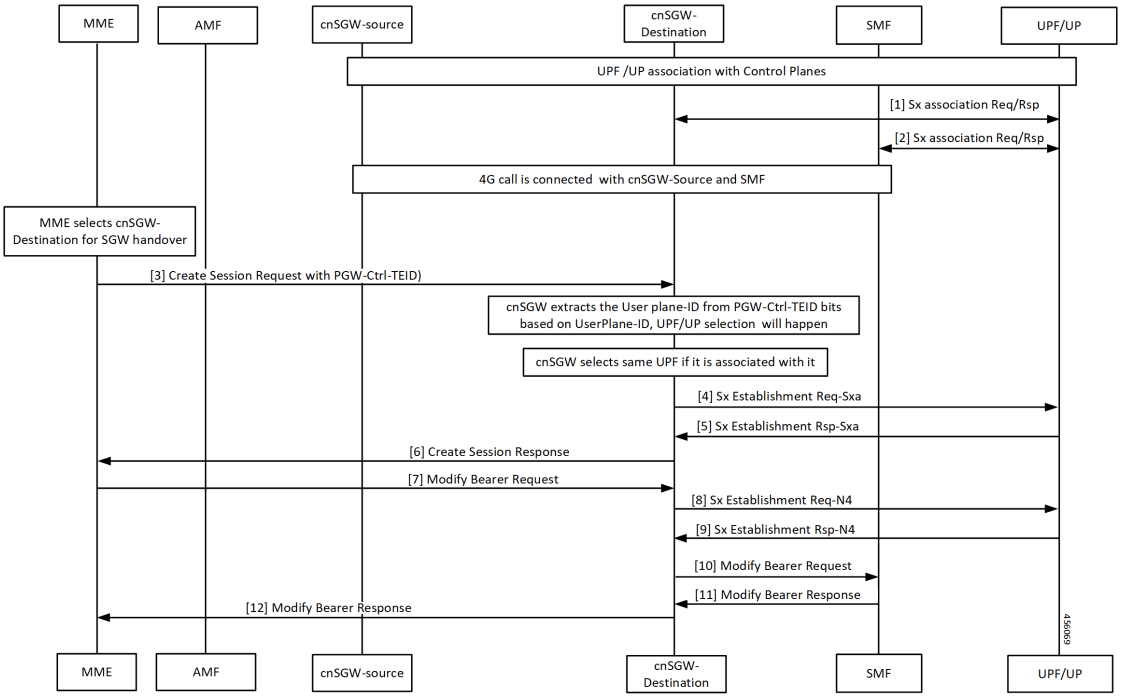


表 *255 : 5G* 対応デバイスの *4G RAT* での *SGW* 間ハンドオーバーコールフローの説明

ステップ	説明
1	コントロールプレーンとの UPF 関連付けの確立 UPF が Sx 関連付け要求/応答を宛先 cnSGW-C に送信します。
2	UPF が Sx 関連付け要求/応答を SMF に送信します。
3	cnSGW-C-Source と SMF 間で 4G コールが接続されました。 MME が SGW ハンドオーバー用に cnSGW-C-Destination を選択します。 MME が PGW-Ctrl-TEID とともに Create Session Req を cnSGW-C-Destination に送信します。
4	cnSGW-C が PGW-Ctrl-TEID から関連する同じ UPF ID を抽出します。 cnSGW-C-Destination が Sx Establishment Req（SXA）を UPF に送信します。
5	cnSGW-C-Destination が UPF から Sx Establishment Rsp（SXA）を受信します。
6	cnSGW-C-Destination が Create Session Response を MME に送信します。
7	cnSGW-C-Destination が MME から Modify Bearer Request を受信します。

ステップ	説明
8	cnSGW-C-Destination が Sx Establishment Req-N4 を UPF に送信します。
9	cnSGW-C-Destination が UPF から Sx Establishment Rsp-N4 を受信します。
10	cnSGW-C-Destination が Modify Bearer Request を SMF に送信します。
11	cnSGW-C-Destination が SMF から Modify Bearer Response を受信します。
12	cnSGW-C-Destination が Modify Bearer Request を MME に転送します。

N26 インターフェイスを使用した 5G から EPS へのハンドオーバー：cnSGW-C および SMF の個別ノードのコールフロー

このセクションでは、N26 インターフェイスを使用した 5G から EPS へのハンドオーバー：cnSGW-C および SMF の個別ノードのコールフローについて説明します。

図 *136 : N26* インターフェイスを使用した *5G* から *EPS* へのハンドオーバー：*cnSGW-C* および *SMF* の個別ノードのコールフロー

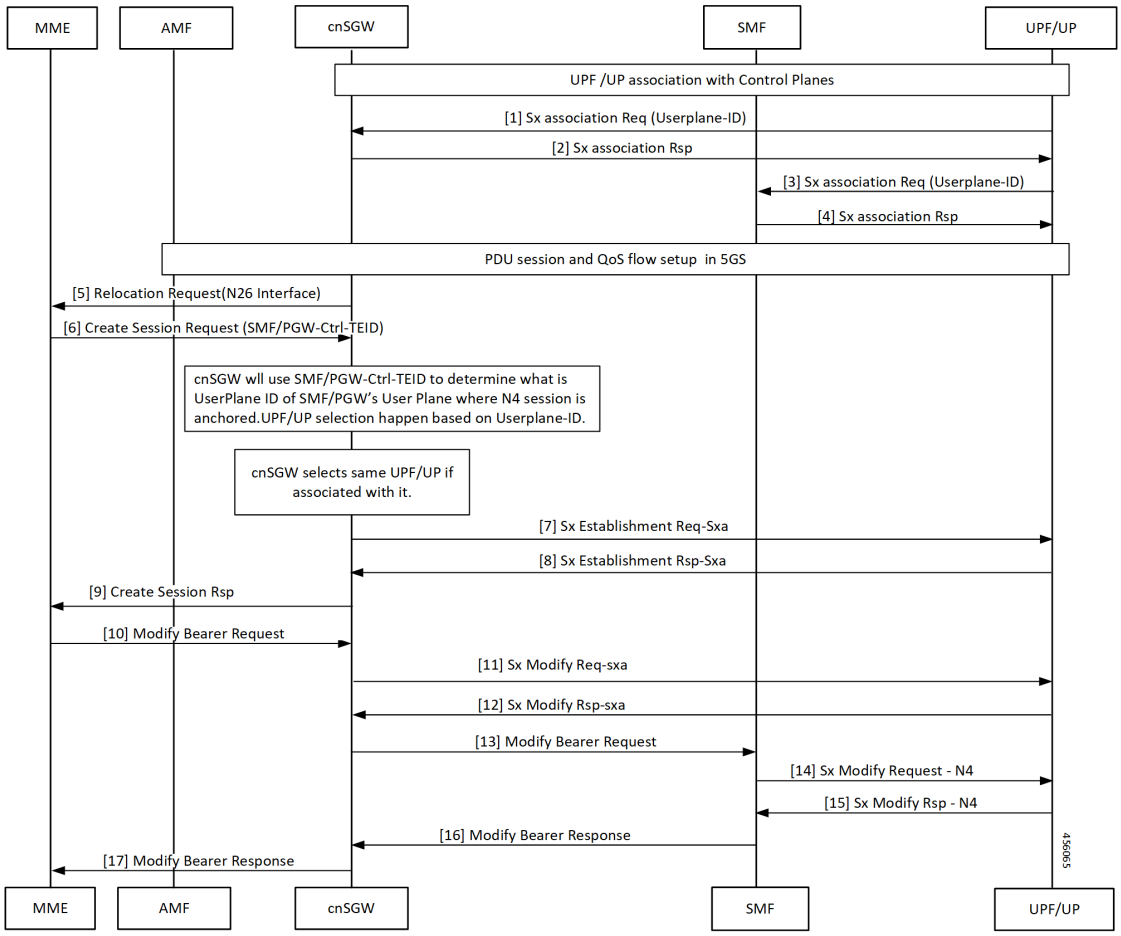


表 *256 : N26* インターフェイスを使用した *5G* から *EPS* へのハンドオーバー：*cnSGW-C* および *SMF* の個別ノードのコールフローの説明

ステップ	説明
1	cnSGW-C が、関連付けられた同じ UPF を選択します。 cnSGW-C が、UPF/UP からユーザープレーン ID を含む Sx 確立要求を受信します。
2	cnSGW-C がSx 確立応答を UPF/UP に送信します。
3	UPF/UP が、ユーザープレーン ID を含む Sx 関連付け要求を SMF に送信します。
4	SMF が Sx 関連付け応答を UPF/UP に送信します。
5	cnSGW-C が、インターフェイス N26 で再配置要求を MME に送信します。
6	MME が、SMF および PGW-ctrl-TEID 情報を含むセッション作成要求を cnSGW-C に送信します。
7	cnSGW-C が、関連付けられた同じ UPF を選択します。 cnSGW-C が Sx 確立要求（SXA）を UPF/UP に送信します。
8	cnSGW-C が UPF/UP から Sx 確立応答（SXA）を受信します。
9	cnSGW-C がセッション作成応答を MME に送信します。
10	MME が、ベアラー変更要求を cnSGW-C に送信します。
11	cnSGW-C が Sx 変更要求（SXA）を UPF/UP に送信します。
12	cnSGW-C が、UPF/UP から Sx 変更応答（SXA）を受信します。
13	cnSGW-C が、ベアラー変更要求を SMF に送信します。
14	SMF が Sx 変更要求（N4）を UPF/UP に送信します。
15	SMF が UPF/UP から Sx 変更応答を受信します。
16	cnSGW-C が、SMF へのベアラー変更応答を受信します。
17	SGW-C が、ベアラー変更応答を MME に転送します。

Wi-Fi から LTE へのハンドオーバーコールフロー

このセクションでは、Wi-Fi から LTE へのハンドオーバーコールフローについて説明します。

図 *137 : Wi-Fi* から *LTE* へのハンドオーバーコールフロー

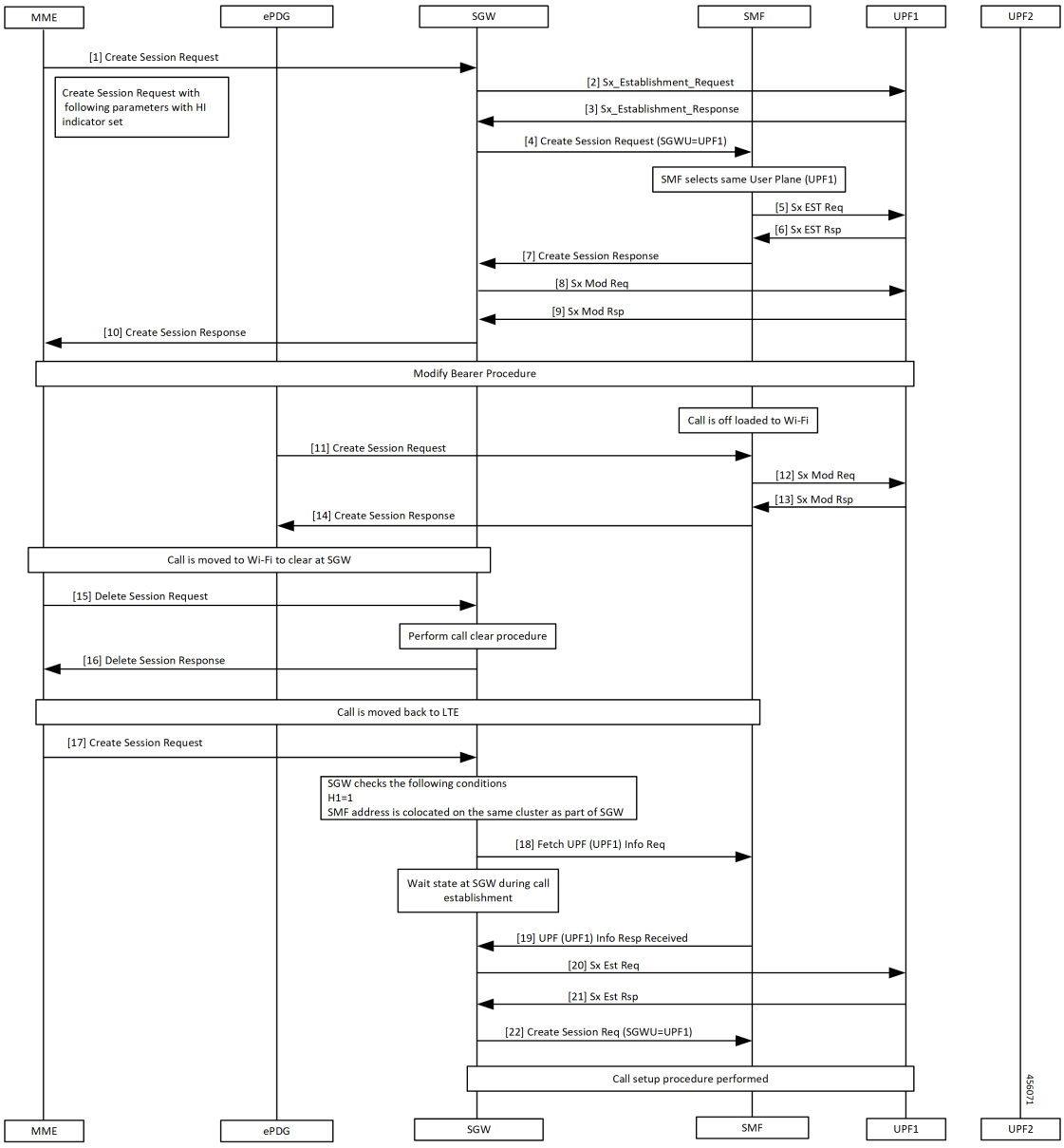


表 *257 : Wi-Fi* から *LTE* へのハンドオーバーコールフローの説明

ステップ	説明
1	MME がセッション作成要求を SGW に送信します。
2	SGW が Sx 確立要求を UPF1 に送信します。
3	SGW が SMF から Sx 確立応答を受信します。

ステップ	説明
4	SGW がセッション作成要求を SMF に送信します。SMF は同じ UPF を選択します。
5	SMF が Sx 確立要求を UPF1 に送信します。
6	SMF が UPF1 から Sx 確立応答を受信します。
7	SMF がセッション作成応答を SGW に送信します。
8	SGW が UPF1 に Sx 変更要求を送信します。
9、10	SGW が UPF1 から Sx 変更応答を受信し、MME に転送します。
11	ベアラー変更手順が行われ、Wi-Fi にコールがオフロードされます。 ePDG が SGW にセッション作成要求を送信します。
12	SMF が Sx 変更要求を UPF1 に送信します。
13	SMF が UPF1 から Sx 変更応答を受信します。
14	ePDG が SGW からセッション作成応答を受信します。
15	コールが Wi-Fi に移動され、SGW でクリアされます。 MME が SGW にセッション削除要求を送信します。
16	コールクリア手順を実行後、MME は SGW からセッション削除応答を受信します。
17	MME がセッション作成要求を SGW に送信します。
18	SGW は、SMF が cnSGW-C と同じクラスタであることを確認した後、UPF（UPF1）情報取得要求を SMF に送信します。
19	SGW が SMF から UPF（UPF1）情報応答を受信します。
20	SGW が Sx 確立要求を UPF1 に送信します。
21	SGW が UPF1 から Sx 確立応答を受信します。
22	SGW が SGWU=UPF1 のセッション作成要求を UPF1 に送信し、コール設定手順を実行します。

cnSGW-C と SMF の複合 UPF 選択の構成

このセクションでは、cnSGW-C と SMF の複合 UPF 選択を構成する方法について説明します。

コンバージド コア プロファイルの構成

ここでは、コンバージドコア プロファイルの構成方法を説明します。

次のコマンドを使用して、UPF 選択が有効になっているプロファイル コンバージドコアを構成します。

```
config
  profile converged-core core_name
    max-upf-index value
    no upf-selection disable
  exit
```

次のコマンドを使用して、UPF 選択が無効になっているプロファイル コンバージドコアを構成します。

```
config
  profile converged-core core_name
    max-upf-index value
    upf-selection disable
  exit
```

注：

- **converged-core** *core_name*：コンバージド コア プロファイル名を指定します。
- **max-upf-index** *value*：UPF インデックス値を指定します。1 ー 1023 の範囲の整数で指定する必要があります。
- **no upf-selection disable**：共存 UPF 選択を有効にします。
- **upf-selection disable**：共存 UPF 選択を無効にします。

設定例

次に、UPF の選択が有効になっている構成例を示します。

```
config
profile converged-core cc1
max-upf-index 1023
no upf-selection disable
exit
```

次に、UPF の選択が無効になっている構成例を示します。

```
config
profile converged-core cc1
max-upf-index 1023
upf-selection disable
exit
```

プロファイル コンバージドコア構成の確認

ここでは、プロファイル コンバージドコア構成を確認する方法を説明します。

次の show コマンドを使用して、UPF 選択が有効になっているプロファイル コンバージドコア設定を表示します。

```
show running-config profile converged-core cc1
profile converged-core cc1
max-upf-index 1023
no upf-selection disable
exit
```

次の show コマンドを使用して、UPF 選択が無効になっているプロファイル コンバージドコア設定を表示します。

```
show running-config profile converged-core cc1
profile converged-core cc1
max-upf-index 1023
upf-selection disable
exit
```

ノード ID の設定

ここでは、ノード ID の設定方法について説明します。

次のコマンドを使用して、ノード ID を設定します。

```
config
  profile network-element upf upf_name
    node-id node_id_value
  exit
```

注：

- **network-element upf** *upf_name*：UPF ピアネットワーク要素名を指定します。
- **node-id** *node_id_value*：UPF ノードのノード ID を指定します。文字列で指定する必要があります。

設定例

次に設定例を示します。

```
config
profile network-element upf upf1
node-id upf1@cn.com
exit
```

UPF 選択 OAM のサポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

UE 切断統計

```
sgw_ue_disconnect_stats{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
 reason="userplane_info_not_available",service_name="sgw-service"} 24
```

PDN 切断統計

```
sgw_pdn_disconnect_stats{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
pdn_type="ipv4",rat_type="EUTRAN",reason="userplane_info_not_available",service_name="sgw-service"}
 8
```

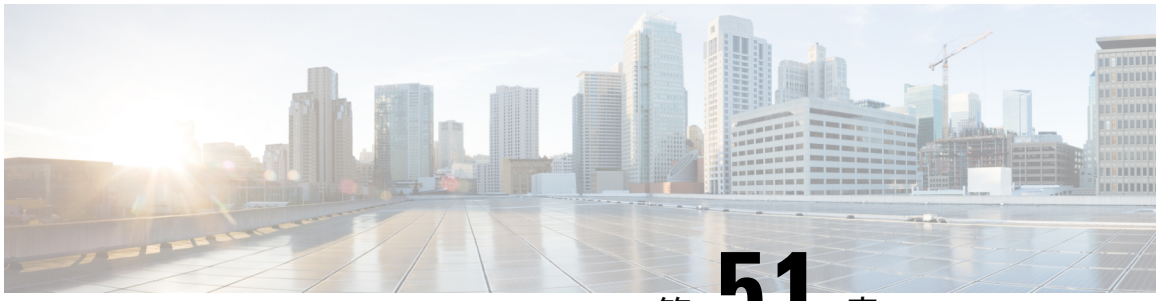
```
sgw_pdn_disconnect_stats{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
pdn_type="ipv4v6",rat_type="EUTRAN",reason="userplane_info_not_available",service_name="sgw-service"}
 15
```

```
sgw_pdn_disconnect_stats{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
pdn_type="ipv6",rat_type="EUTRAN",reason="userplane_info_not_available",service_name="sgw-service"}
 1
```

SGW サービス統計

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="userplane_selection_fail",
instance_id="0",interface="interface_sgw_ingress",reject_cause="no_resources_available",service_name=
"sgw-service",sgw_procedure_type="initial_attach",status="failure",sub_fail_reason=""}
 22
```

```
sgw_service_stats{app_name="smf",cluster="cn",data_center="cn",fail_reason="userplane_selection_fail",
instance_id="0",interface="interface_sgw_ingress",reject_cause="no_resources_available",service_name="sgw-service",
 sgw_procedure_type="secondary_pdn_creation",status="failure",sub_fail_reason=""}
 2
```



第 51 章

VoLTE コールの優先順位付け

- 機能の概要と変更履歴 （797 ページ）
- 機能説明 （798 ページ）
- 機能の仕組み （798 ページ）
- 機能設定 （798 ページ）
- OAM サポート （801 ページ）

機能の概要と変更履歴

要約データ

表 *258* : 要約データ

該当製品または機能エリア	cnSGW-C
該当プラットフォーム	SMI
機能のデフォルト設定	有効、常時オン
関連資料	該当なし

マニュアルの変更履歴

表 *259* : マニュアルの変更履歴

改訂の詳細	リリース
拡張および非標準（オペレータ定義）の QCI 値のサポートが追加されました。	2021.02.3
最初の導入。	2021.02.0

機能説明

cnSGW-C は以下を提供します。

- QCI を IMS メディアとしてマークするための CLI サポート。
- `show subscriber` の出力で、セッション/ベアラーが VoLTE かどうかを表示する CLI サポート。
- システム内の VoLTE サブスクライバの数を識別するカウンタのサポート。
- VoLTE でマークされたセッションに基づく Sx メッセージの優先順位の設定。

機能の仕組み

ここでは、この機能の仕組みを説明します。

- SGW プロファイルは SGW サービスを表します。
- SGW プロファイルには、オペレータポリシーの選択に役立つサブスクライバポリシーが関連付けられています。
- オペレータポリシーにはDNNポリシーが関連付けられています。
- DNN ポリシーには、VoLTE サブスクライバの優先順位をマークするための QCI マークを持つ DNN プロファイルが関連付けられています。

`show subscriber` コマンドを実行すると、IMS としての QCI マーキングに基づいて、*volteBearer* および *volteSession* フラグが内部的に設定されます。

- *volteBearer* はベアラーレベルのフラグです。ベアラー QCI がマークされた QCI リストに存在する場合、*volteBearer* フラグは true に設定され、ベアラーは **volteBearer** と見なされます。
- *volteSession* はセッションレベルのフラグです。そのサブスクライバのいずれかの PDN に VoLTE ベアラーが存在する場合、このフラグは **true** に設定されます。

機能設定

この機能の設定には、次の手順が含まれます。

- コール優先順位を設定します。詳細については、優先順位の構成 （799 ページ） を参照してください。
- メッセージの優先順位を設定します。詳細については、Sx メッセージの優先順位 （801 ページ）を参照してください。

優先順位の構成

ここでは、優先順位を構成する方法について説明します。

DNN プロファイルで QCI レベルを VoLTE メディアとしてマークするには、CLI を使用します。コールで要求された QCI がマークされた QCI と一致する場合、SGW は *volteSession* および *volteBearer* フラグを設定します。サブスクライバセッションに **volteSession** がある場合は、そのサブスクライバの優先順位は他のサブスクライバと比較して最も高くなります。

```
config
   profile dnn profile_name ims mark qci qci_value
```

注：

- **profile dnn** *profile_name*：DNN プロファイル名を指定します。
- **mark**：標準 QCI 値を IMS メディアとしてマークします。
- **qci** *qci_value*：QCI 値を指定します。次の QoS クラス識別子がサポートされています。

  標準：1 ～ 9

  拡張：65、66、69、70、80、82、83

  非標準（オペレータ定義）：128 ～ 254

設定例

次に、設定例を示します。

```
config
   profile dnn dnn1 ims mark qci [ 2 3 4 ]
   end
```

設定の確認

設定を確認する方法

```
show full-configuration profile dnn dnn1
profile dnn dnn1
 ims mark qci [ 2 3 4 ]
```

IMS としての QCI マーキングに基づいて、`show subscriber` コマンドを実行すると、volteSession および volteBearer フラグが内部的に設定されます。

ここでは、出力例を示します。

```
show subscriber namespace sgw imsi 123456789012348
subscriber-details
{
  "subResponses": [
    {
      "status": true,
      "genericInfo": {
        "imsi": "imsi-123456789012348",
        "mei": "imei-123456786666660",
        "msisdn": "msisdn-223310101010101",
```

```
  "accessType": "EUTRAN",
  "plmnId": {
    "mcc": "123",
    "mnc": "456"
  },
  "sgwProfileName": "sgw1",
  "unAuthenticatedImsi": "No"
},
"s11cInterfaceInfo": {
  "sgwTeid": "[0x12000147] 301990215",
  "sgwIPv4Address": "209.165.201.19",
  "mmeTeid": "[0x62b5] 25269",
  "mmeIPv4Address": "209.165.201.20"
},
"pdnInfoList": {
  "totalPdn": 1,
  "pdnInfo": [
    {
      "pdnId": "PDN-1",
      "apn": "intershat",
      "attachType": "Initial Attach",
      "sgwRelocState": "N/A",
      "operatorPolicyName": "N/A",
      "dnnProfileName": "N/A",
      "defaultEbi": 5,
      "pdnType": "IPv4",
      "allocatedIPv4": "209.165.201.26",
      "apnSelectionMode": "Subscribed",
      "ambrUplink": "10 Kbps",
      "ambrDownlink": "20 Kbps",
      "s5cInterfaceInfo": {
        "sgwTeid": "[0x52000147] 1375732039",
        "sgwIPv4Address": "209.165.201.19",
        "pgwTeid": "[0x339a] 13210",
        "pgwIPv4Address": "209.165.201.18"
      },
      "sxaInterfaceInfo": {
        "selectedUP": "209.165.201.20",
        "upEpKey": "209.165.201.20:209.165.201.19",
        "cpSeid": "[0x12000147520000147] 1297038098512740679",
        "upSeid": "[0x2712] 10002"
      },
      "bearerInfoList": {
        "totalBearer": 1,
        "bearerInfo": [
          {
            "bearerId": "Bearer-1",
            "state": "Connected",
            "ebi": 5,
            "isDefaultBearer": true,
            "qosInfo": {
              "qci": 6,
              "arp": 113
            },
            "sluInterfaceInfo": {
              "sgwTeid": "[0x62b7] 25271",
              "sgwIPv4Address": "209.165.200.226",
              "eNodeBTeid": "[0x62b8] 25272",
              "eNodeBIPv4Address": "209.165.201.20"
            },
            "s5uInterfaceInfo": {
              "sgwTeid": "[0x62b6] 25270",
              "sgwIPv4Address": "209.165.201.1",
              "pgwTeid": "[0x339b] 13211",
```

```
                "pgwIPv4Address": "209.165.201.18"
              },
              "chargingId": 303174163
            }
          ]
        },
        "uli": {
          "mcc": "123",
          "mnc": "456",
          "tac": "0x92a",
          "eci": "0x12d687"
        },
        "uetimeZone": {
          "timeZone": "+0:15",
          "dayLightSavingTime": "+1 hour"
        },
        "plmnType": "VISITOR"
      }
    ]
  }
 }
]
}
```

Sx メッセージの優先順位

ここでは、Sx メッセージの優先順位について説明します。

VoLTE フラグ（volteSession および volteBearer）に基づいて、SGW は受信した要求/応答（S5-CBReq、S11CBResp、 S11-UBRes、S5DBReq など）を処理しながら、Sx 要求メッセージ（Sx-Est、Sx-Mod など）にメッセージの優先順位を設定します。

セッションに volteSession（または VoLTE QCI）のマークが付いている場合、Sx メッセージの優先順位が設定されます。

Sx 要求（確立または変更）メッセージヘッダー内：

```
"HEADERS": {
      "length": 30,
      "msg_type": 52,
      "priority": 2,        <<< VOLTE Session priority value
      "priority_flag": 1,   <<< Priority flag is true i.e 1
      "retransmit": 0,
      "seid": 10002,
      "seid_flag": 1,
      "seq_number": 22,
      "version": 1
    },
```

OAM サポート

ここでは、この機能の操作、管理、およびメンテナンスに関して説明します。

バルク統計

sgw_voltesession_counter は、*volteSession* フラグに基づいて追加されます。このカウンタは、特定の時点でシステムに存在する VoLTE セッションの数を示します。

volteSession フラグが true の場合、カウンタが増分されます。

volteSession フラグが存在しない場合（VoLTE ベアラーが PDN に存在しない場合）、カウンタは減少します。

カウンタ名： sgw_volsession_counter

説明：システムに存在する現在アクティブな VoLTE セッション。

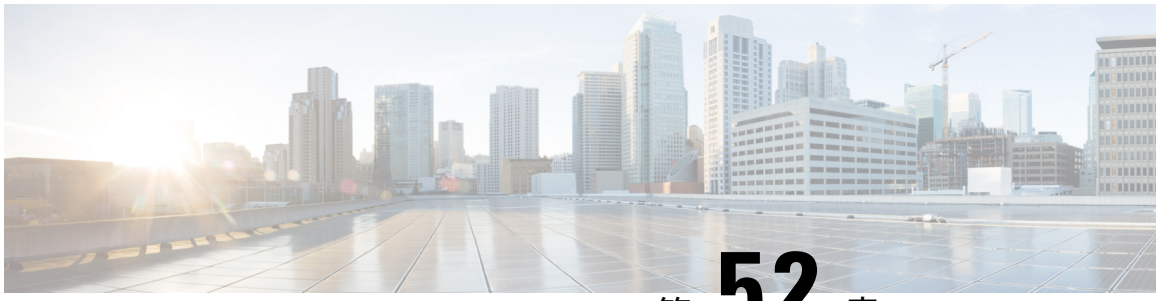
Label:

- LABEL_STAT：VolteSession

サンプルカウンタの出力：

ポッドからカウンタを確認できます。

```
curl http://209.165.201.20:8080/metrics | grep "volte"

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0#
HELP sgw_voltesession_counter Current Active Volte Session
# TYPE sgw_voltesession_counter gauge
sgw_voltesession_counter{app_name="smf",cluster="cn",data_center="cn",instance_id="0",
service_name="sgw-service",state="VolteSession"} 1
100  246k    0  246k    0     0  16.0M      0 --:--:-- --:--:-- --:--:-- 17.1M
root@sgw-service-n0-0:/opt/workspace#
```



第 52 章

cnSGW-C のトラブルシューティング

- 説明 （803 ページ）
- CLI データの使用 （803 ページ）
- ログ （808 ページ）

説明

この章では、システムの動作中に発生する可能性のある問題をトラブルシューティングするために、コマンドラインインターフェイス（CLI）コマンドとログの使用に関する情報について説明します。

CLI データの使用

このセクションでは、トラブルシューティングに使用する show コマンドと clear コマンド、および monitor コマンドについて説明します。

show subscriber および cdl show コマンド

表 *260 :* 機能の履歴

機能名	リリース情報	説明
Show および Clear Subscriber CLI コマンドによる IPv6 専用 eNB 挿入のサポート	2024.02.0	ネットワークに IPv6 専用の eNB を追加する前に、IPv4 専用 eNB セッションから IPv6 専用 eNB セッションへのハンドオーバーを成功させるために、メッシュ内のすべての UPF で IPv6 を有効にする必要があります。さらに、V6 専用 eNB を挿入する前に、すべてのセッションに V4V6 トンネルが必要です。この IPV6 専用 eNB 挿入をサポートするために、cnSGW-c には次の CLI コマンドが含まれています。 • **show subscriber nf-service sgw data-tunnel** *data_tunnel_type* および **show subscriber count nf-service sgw data-tunnel** *data_tunnel_type* CLI コマンド • **clear subscriber nf-service sgw data-tunnel** *data_tunnel_type* CLI コマンド **デフォルト設定**：該当なし

ここでは、トラブルシューティング情報について説明します。

- SGW サブスクライバ情報を表示するには、次のコマンドを使用します。

  show subscriber namespace sgw imsi *imsi_value*

  show subscriber nf-service sgw imsi *imsi_value*

  show subscriber count { all }

```
show subscriber namespace sgw imsi 123456789012348
subscriber-details
{
  "subResponses": [
    {
      "status": true,
```

```
"genericInfo": {
  "imsi": "imsi-123456789012348",
  "mei": "imei-123456786666660",
  "msisdn": "msisdn-223310101010101",
  "accessType": "EUTRAN",
  "plmnId": {
    "mcc": "123",
    "mnc": "456"
  },
  "sgwProfileName": "sgw1",
  "unAuthenticatedImsi": "No"
},
"s11cInterfaceInfo": {
  "sgwTeid": "[0x12000147] 301990215",
  "sgwIPv4Address": "209.165.201.19",
  "mmeTeid": "[0x62b5] 25269",
  "mmeIPv4Address": "209.165.201.20"
},
"pdnInfoList": {
  "totalPdn": 1,
  "pdnInfo": [
    {
      "pdnId": "PDN-1",
      "apn": "intershat",
      "attachType": "Initial Attach",
      "sgwRelocState": "N/A",
      "operatorPolicyName": "N/A",
      "dnnProfileName": "N/A",
      "defaultEbi": 5,
      "pdnType": "IPv4",
      "allocatedIPv4": "209.165.201.26",
      "apnSelectionMode": "Subscribed",
      "ambrUplink": "10 Kbps",
      "ambrDownlink": "20 Kbps",
      "s5cInterfaceInfo": {
        "sgwTeid": "[0x52000147] 1375732039",
        "sgwIPv4Address": "209.165.201.19",
        "pgwTeid": "[0x339a] 13210",
        "pgwIPv4Address": "209.165.201.18"
      },
      "sxaInterfaceInfo": {
        "selectedUP": "209.165.201.20",
        "upEpKey": "209.165.201.20:209.165.201.19",
        "cpSeid": "[0x1200014752000147] 1297038098512740679",
        "upSeid": "[0x2712] 10002"
      },
      "bearerInfoList": {
        "totalBearer": 1,
        "bearerInfo": [
          {
            "bearerId": "Bearer-1",
            "state": "Connected",
            "ebi": 5,
            "isDefaultBearer": true,
            "qosInfo": {
              "qci": 6,
              "arp": 113
            },
            "s1uInterfaceInfo": {
              "sgwTeid": "[0x62b7] 25271",
              "sgwIPv4Address": "209.165.200.226",
              "eNodeBTeid": "[0x62b8] 25272",
              "eNodeBIPv4Address": "209.165.201.20"
            },
```

```
                    "s5uInterfaceInfo": {
                      "sgwTeid": "[0x62b6] 25270",
                      "sgwIPv4Address": "209.165.201.1",
                      "pgwTeid": "[0x339b] 13211",
                      "pgwIPv4Address": "209.165.201.18"
                    },
                    "chargingId": 303174163
                  }
                ]
              },
              "uli": {
                "mcc": "123",
                "mnc": "456",
                "tac": "0x92a",
                "eci": "0x12d687"
              },
              "uetimeZone": {
                "timeZone": "+0:15",
                "dayLightSavingTime": "+1 hour"
              },
              "plmnType": "VISITOR"
              "dualStackEnabled": true,
              "dataTunnelType": "IPV4"
              "ipSrcViolation": "N/A",
              "mmeS4SgsnId": {
                "IPv4": "10.10.10.1"
              }
            }
          ]
        }
      }
    ]
}
```


```
show subscriber count all
subscriber-details
{
   "sessionCount": 50
}
```

- セッションの概要情報を表示するには、次のコマンドを使用します。

  cdl show sessions summary

```
cdl show sessions summary
message params: {session-summary cli session {0 100 0 [] 0 0}}
session {
    primary-key imsi-146062234105885
    unique-key [ 16777218 ]
    map-id 1
    instance-id 1
    version 1
    create-time 2020-04-27 16:18:24.225646626 +0000 UTC
    last-updated-time 2020-04-27 16:18:24.87241245 +0000 UTC
    purge-on-eval false
    next-eval-time 2020-05-04 16:18:24 +0000 UTC
    data-size 406
}
```

- データトンネルタイプに基づいてサブスクライバの出力を表示するには、次のコマンドを使用します。

  show subscriber nf-service sgw data-tunnel *data_tunnel_type*

```
show subscriber nf-service sgw data-tunnel IPV4V6
subscriber-details
{
  "subResponses": [
    [
      "id-index:1:0:32768",
      "id-index-key:1:0:globalKey:32768",
      "id-value:16777344",
      "imsi:imsi-123456789012348",
      "msisdn:msisdn-223310101010101",
      "imei:imei-123456786666660",
      "upf:10.1.12.212",
      "upfEpKey:10.1.12.212:10.1.10.78",
      "subscribertype:non-volte",
      "s5s8Ipv4:10.1.15.100",
      "s11Ipv4:10.1.12.212",
      "data-tunnel:IPV4V6",
      "namespace:sgw",
      "nf-service:sgw"
    ]
  ]
}
```

- データトンネルタイプに基づいてサブスクライバ数の出力を表示するには、次のコマンドを使用します。

  show subscriber count nf-service sgw data-tunnel *data_tunnel_type*

```
show subscriber count nf-service sgw data-tunnel IPV4V6
subscriber-details
{
  "sessionCount": 1
}
```

- サブスクライバ情報をクリアするには、次のコマンドを使用します。

  clear subscriber all

  clear subscriber nf-service sgw all

```
clear subscriber all
result
ClearSubscriber Request submitted
```

```
clear subscriber nf-service sgw all
result
ClearSubscriber Request submitted
```

- データトンネルタイプを使用してサブスクライバ情報をクリアするには、次のコマンドを使用します。

  clear subscriber nf-service sgw data-tunnel *data_tunnel_type*

```
clear subscriber nf-service sgw data-tunnel IPV4V6
result
Clear subscriber request submitted successfully for GR Instance ID 1. Waiting Time
is 9.351706 seconds
```

ログ

システムロギング機能は、アプリケーション間でログメッセージをログに記録する共通の方法を提供します。各ログは次のコンポーネントで構成されています。

- タイムスタンプ：ログ作成の日時を示します。
- ログメッセージ：特定のログのメッセージを示します。
- ログレベル：ログ メッセージの重要性レベルを示します。
- ログタグ：モジュール名、コンポーネント名、およびインターフェイス名の詳細を表示します。ログタグは事前に作成され、ロギング中に渡されます。

イベント障害のログ

表 *261 :* 機能の履歴

機能名	リリース情報	説明
サービスポッドのイベント障害ログ	2025.01.0	この機能を使用すると、一貫性のあるイベント障害ログが拡張され、サービスポッドの Delete Bearer Command、Change Notification、および PDN Disconnect の手順がサポートされます。
サービスポッドのイベント障害ログ	2024.04.0	この機能を使用すると、一貫性のあるイベント障害ログが拡張され、サービスポッドの Create Bearer、Update Bearer、Delete Bearer、PDN Modify List、および Modify Bearer Command の手順がサポートされます。

機能名	リリース情報	説明
イベント障害ログ	2024.03.0	

機能名	リリース情報	説明
		cnSGWc は次のサポートを提供しています。 • ポッド間の PDN Setup、Idle または Active、PDN Modify、および PDN Disconnect の手順に関する一貫したイベント障害ログ • ポッドタイプで設定可能なログ • 要求と応答の詳細を 1 行形式で含める 不要なシステムログが大量に生成されると、メモリ消費が増加し、パフォーマンスが悪化し、ログの管理と使用が非効率的になります。これらの問題を防ぐために、さまざまなポッド間で一貫したエラーログメッセージ形式が導入され、メモリ消費量の削減、システム生成ログ数の最小化、および効率的なトラブルシューティングが可能になりました。メッセージは 1 行のログ形式で表示されるため、可読性が高くなります。 不要なシステムログが大量に生成されると、メモリ消費が増加し、パフォーマンスが悪化し、ログの管理と使用が非効率的になります。これらの問題を防ぐために、さまざまなポッドで一貫したエラーログメッセージ形式が導入されました。cnSGWc 手順の拡張エラーロギングにより、メモリ使用量の削減、システム生成ログ数の最小化、詳細で一貫性のある設定可能なロギング

機能名	リリース情報	説明
		が実現され、効果的なデバッグやシステムモニタリングに役立ちます。 **デフォルト設定**： 該当なし

cnSGWc 手順のエラーロギング機能が強化され、詳細で一貫性のある設定可能なロギングが提供されます。これらの機能拡張は、効果的なデバッグとシステムモニタリングに役立ちます。拡張機能は次のとおりです。

- 一貫したログ形式：単一行のログ形式は、ログの記録および解釈方法を統一するために、RESTエンドポイントやサービスポッドなど、さまざまなポッド間で標準化されています。
- ロギングの有効化と無効化：ロギングを有効または無効にするオプションは、特定のポッドタイプで使用できます。このオプションを使用すると、ログストレージを柔軟に管理できます。
- 詳細なログコンテンツ：ログには、プライマリキー、インターフェイス、手順の詳細、メッセージ要求と応答などの包括的な詳細が含まれます。このレベルの詳細は、デバッグと分析の詳細に役立ちます。
- ログレベル管理：デフォルトでは、ログは INFO レベルで書き込まれます。必要に応じて、ログを有効にできます。ログを有効にすると、生成および保存されるログデータの量を制御できます。

  デフォルトでは、ログレベルは WARN に設定され、ロギングはデフォルトで無効になります。必要に応じて、ロギングを有効にできます。
- 共通ロギングインターフェイス：イベントロギング用に共通のインターフェイスが実装されます。すべてのコンポーネントで、同じロギング標準と形式に対してこのインターフェイスが使用されます。
- ログタグ：ログタグが強化され、特定のポッドタイプまたはサービスのログを有効または無効にし、ロギングをきめ細かく制御できるようになりました。
- サポートされるログ：ロギングは、さまざまな手順、ポッド、およびインターフェイスでサポートされています。

機能の仕組み

各ポッド全体で一貫したログ形式を持つために、各コンポーネントはイベントロギングに共通のインターフェイスを使用します。ログは JSON 形式を使用するため、すべてのデータが 1 行に表示されます。ログは INFO レベルで書き込まれるため、デフォルトではこのレベルを無効にして、必要な場合に有効にできます。

ログタグの形式は次のとおりです。

transaction.event.<pod-type>。**<pod-type>** は、ポッドが使用するサービス名です。たとえば、rest-ep および sgw-service です。

ログタグを使用すると、特定のポッドタイプまたはサービスのログを有効または無効にできます。

ログは、対応するログレベルが INFO、DEBUG、または TRACE に設定されている場合に書き込まれます。ただし、メッセージの要求または応答フィールドは、ログレベルが DEBUG または TRACE に設定されている場合にのみ入力されます。

Debug レベルが有効になっている場合のサンプルログ


```
sgw-service-0 [INFO] [Transaction.go:1595] [transaction.event.sgw-service]
 {"TxnId":16,"StartTime":"2024-05-10T15:01:00+05:30","GRInstanceId":1,
"TxnType":"S5CreateSessReq","Priority":33,"SessionNamespace":"sgw(2)",
"CdlSliceName":"1","SubscriberId":"imsi-430967582185910","SessionPrimaryKey":
"imsi-430967582185910","SessionKeys":"imsi-430967582185910 (pk)
subscribertype:wps (nuk) 16777217 (uk) id-index:1:0:32768 (nuk)
id-index-key:1:0:globalKey:32768 (nuk) id-value:16777217 (nuk) imsi:
imsi-430967582185910 (nuk) msisdn:msisdn-9326737733 (nuk) imei:imeisv-1122334455667788
 (nuk) upf:192.168.56.20 (nuk) upfEpKey:192.168.56.20:192.168.56.10
(nuk) s11Ipv4:192.168.56.20 (nuk) s5s8Ipv4:192.168.56.30 (nuk)",
"SessionState":"Create_Session","ErrorMessage":"{\"ErrType\":3,\"ErrCause\":
{\"Value\":89,\"Pce\":false,\"Bce\":false,\"OrigInd\":false,\"OffendingIe\":
{\"Valid\":false,\"Tag\":0,\"Instance\":0,\"Value\":\"\"}},\"BrCtxtCause\":true,
\"SubfailReason\":89,\"SubfailReasonDetailed\":0,\"SubfailStr\":\
"S5 Create Session Response Failure\",\"SubfailReasonStr\":\"IPv4:192.168.56.30
IPv6:\",\"SourceDetails\":\"/opt/workspace/sgw_service/src/sgw-service/procedures
/pdnsetup/idlestate.go:963\"}","MessageRequest":{"Version":2,"TEIDflag":true,
"MsgPriority":true,"MsgTypeId":32,"MsgPriorityValue":10,"meta_data":{"from_ip"
:3232249867,"to_ip":3232249886,"to_port":2123,"intfType":3,"s5edscp":{"Value":12,
"Valid":true},"peerType":1},"MsgType":{"Create_Session_Request":{"IMSI":
"430967582185910","APN":"starent.com","AMBR":{"UL":119,"DL":135},"MEI":
"1122334455667788","MSISDN":"9326737733","Indication":{},"PAA":{"PDN_Type":1,
"IPv4":"0.0.0.0"},"RAT_Type":{"value":6},"Serving_Network":{"MCC":"123","MNC":
"765"},"ULI":{"uli_tai":{"mcc":"214","mnc":"365","val":4660}},"FQ_TEID":{"sgwCntrl":
{"IFace":6,"TEID":1358954497,"IPv4":"192.168.56.11"}},"Bearer_Context_List":
{"num_bearer_ctxt":1,"pbBearerCxt":[{"linked_ebi":{"value":5},"fqteid":{"sgwData":
{"TEID":1073807935,"IPv4":"192.168.56.60","IPv6":"4123::192:168:56:60"}},
"bearerQos":{"PL":2,"QCI":5,"arp":8}}]},"Charging_Characteristics":{"value":
"NBIAAA=="},"PDN_Type":{"value":1},"APN_Restriction":{},"Selection_Mode":
{"value":1}}}},"MessageResponse":{"Version":2,"TEIDflag":true,"MsgPriority":
true,"MsgLength":19,"TIED":1358954497,"Seq":1,"MsgTypeId":33,"MsgPriorityValue":10,
"meta_data":{"from_ip":3232249886,"to_ip":3232249867,"intfType":3},"MsgType":
{"Create_Session_Response":{"Cause":{"Cause_Value":89},"Recovery":{"value":10}}}}}
```


Info レベルが有効になっている場合のサンプルログ


```
sgw-service-0 [INFO] [Transaction.go:1607] [transaction.event.sgw-service]
 {"TxnId":7,"StartTime":"2024-06-20T12:10:05+05:30","GRInstanceId":1,"TxnType":
"S5CreateSessReq","Priority":33,"SessionNamespace":"sgw(2)","CdlSliceName":"1",
"SubscriberId":"imsi-430967582185910","SessionPrimaryKey":"imsi-430967582185910",
"SessionKeys":"imsi-430967582185910 (pk) subscribertype:wps (nuk) 16777217 (uk)
id-index:1:0:32768 (nuk) id-index-key:1:0:globalKey:32768 (nuk) id-value:16777217
(nuk) imsi:imsi-430967582185910 (nuk) msisdn:msisdn-9326737733 (nuk) imei:
imeisv-1122334455667788 (nuk) upf:192.168.56.20 (nuk)
upfEpKey:192.168.56.20:192.168.56.10 (nuk) s11Ipv4:192.168.56.20 (nuk)
s5s8Ipv4:192.168.56.30 (nuk)","SessionState":"Create_Session","ErrorMessage":
"{\"ErrType\":3,\"ErrCause\":{\"Value\":89,\"Pce\":false,\"Bce\":false,\"OrigInd\"
:false,\"OffendingIe\":{\"Valid\":false,\"Tag\":0,\"Instance\":0,\"Value\":\"\"}},
\"BrCtxtCause\":true,\"SubfailReason\":89,\"SubfailReasonDetailed\":0,\"SubfailStr\":
\"S5 Create Session Response Failure\",\"SubfailReasonStr\":\"IPv4:192.168.56.30
IPv6:\",\"SourceDetails\":\"/opt/workspace/sgw_service/src/sgw-service/procedures
/pdnsetup/idlestate.go:971\"}"}
```

サポートされているログ

この機能は、cnSGWc のログをサポートしており、さまざまな手順、ポッド、およびインターフェイス全体の詳細なエラー情報を提供します。次の表に、サポートされているログの一覧を示します。

表 *262 :* サポートされているログ

サポートされているロギングを使用する **cnSGWc** 手順	ロギングに関係するポッド	ポッドと関連付けられている **cnSGWc** インターフェイスと手順
PDN Setup	SERVICE	S11、S5、SXA、RMGR
Idle/Active	SERVICE	S11、S5、SXA
PDN Modify	SERVICE	S11、SXA、S5
PDN Disconnect（DSR/DBR）	SERVICE	S11、S5、SXA、RMGR
Create Bearer	SERVICE	S11、S5、SXA
Update Bearer	SERVICE	S11、S5、SXA
Delete Bearer（専用）	SERVICE	S11、S5、SXA
PDN Modify List	SERVICE	S11、S5、SXA
ベアラー変更コマンド	SERVICE	S11、S5、SXA
ベアラー削除コマンド	SERVICE	S11、S5、SXA
Change Notification	SERVICE	S11、S5、SXA
PDN Disconnect	SERVICE	S11、S5、SXA、RMGR

イベントロギングの有効化

cnSGWc サービスのイベントログを有効または無効にするには、次の手順を使用します。CLI コマンドを使用して適切なログレベル構成を行うことで、生成されるログデータの量とタイプを制御できます。したがって、この構成は効果的なモニタリング、トラブルシューティング、およびパフォーマンス管理に役立ちます。

手順

ステップ 1 イベントロギングのログレベルを指定します。

logging name transaction.event.<*podname*> **level application** *application_log_level*

<pod-type> は、ポッドが使用するサービス名です。

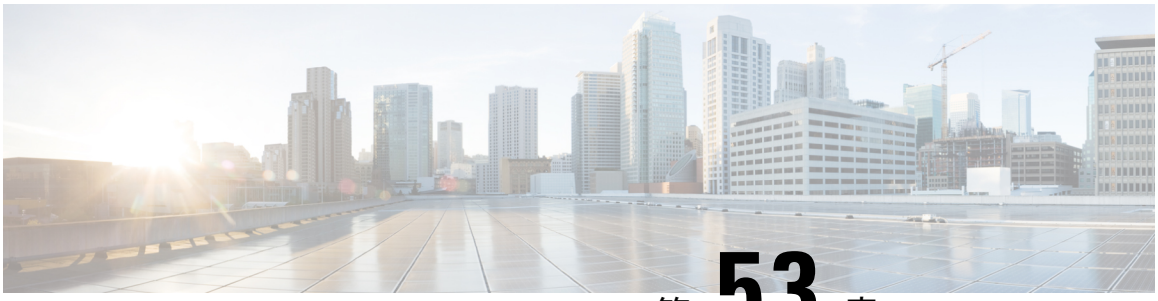
例：

logging name transaction.event.*sgw-service* **level application** *[debug | error | info | off | trace | warn]*

ステップ 2 （オプション）**show running-config logging** コマンドを使用して、構成されたアプリケーション イベント ロギング レベルを確認します。

例：

```
show running-config logging
   logging level tracing debug
   logging name infra.config.core level application trace
   logging name infra.config.core level transaction trace
   logging name infra.config.core level tracing off
   logging name infra.message_log.core level transaction trace
   logging name transaction.event.sgw-service level application debug
```



第 53 章

cnSGW-C 構成例

- 設定例 （815 ページ）

設定例

以下に設定例を示します。

```
show running-config
profile compliance comp1
 service nsmf-pdusession
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service namf-comm
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service n1
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service n2
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service nudm-sdm
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service nudm-uecm
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service nnrf-disc
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service nnrf-nfm
```

```
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service npcf-smpolicycontrol
  version uri v1
  version full 1.0.0
  version spec 15.4.0
 exit
 service nchf-convergedcharging
  version uri v1
  version full 1.0.0
  version spec 15.3.0
 exit
exit
profile network-element amf amf1
 nf-client-profile        AP1
 failure-handling-profile FH3
 query-params [ dnn ]
exit
profile network-element udm udm1
 nf-client-profile        UP1
 failure-handling-profile FH4
 query-params [ dnn ]
exit
profile network-element pcf pcf1
 nf-client-profile        PP1
 failure-handling-profile FH1
 query-params [ dnn ]
 rulebase-prefix          cbn#
 predefined-rule-prefix   crn#
exit
profile network-element chf chf1
 nf-client-profile         CP1
 failure-handling-profile  FH2
 query-params [ dnn ]
 nf-client-profile-offline CP2
exit
profile network-element chf chgser1
exit
profile network-element upf upf1
 node-id      upf1@sgw.com
 n4-peer-address ipv4 209.165.200.234
 n4-peer-port 8805
 dnn-list     [ cisco.com intershat starent.com ]
 capacity     65535
 priority     65535
exit
profile upf-group group1
 failure-profile FH1
exit
profile icmpv6 icmpprf1
 options virtual-mac b6:6d:57:45:45:45
exit
profile charging chgprf1
 method [ offline ]
exit
profile charging-characteristics 1
 charging-profile chgprf1
exit
profile failure-handling FH1
 interface pfcp
  message N4SessionEstablishmentReq
   cause-code pfcp-entity-in-congestion action retry-terminate max-retry 2
```

```
    cause-code system-failure action terminate
    cause-code service-not-supported action terminate
    cause-code no-resource-available action retry-terminate max-retry 3
    cause-code no-response-received action retry-terminate max-retry 1
    cause-code reject action terminate
   exit
   message N4SessionModificationReq
    cause-code mandatory-ie-incorrect action terminate
    cause-code session-ctx-not-found action terminate
    cause-code reject action terminate
   exit
  exit
 exit
profile failure-handling gtp1
 interface gtpc message S5S8CreateBearerReq
  cause-code temp-fail
   action retry timeout 1000 max-retry 2
  exit
 exit
 interface gtpc message S5S8UpdateBearerReq
  cause-code temp-fail
   action retry timeout 1000 max-retry 2
  exit
 exit
 interface gtpc message S5S8DeleteBearerReq
  cause-code temp-fail
   action retry timeout 1000 max-retry 2
  exit
 exit
exit
profile access access1
 n26 idft enable timeout 15
 n2 idft enable timeout 15
 gtpc gtpc-failure-profile gtp1
exit
profile dnn default-profile
 network-element-profiles chf chf1
 network-element-profiles amf amf1
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 charging-profile chgprf1
 virtual-mac      b6:6d:47:47:47:47
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV6 IPV4V6 ]
 upf apn starent.com
exit
profile dnn intershat
 network-element-profiles chf chf1
 network-element-profiles amf amf1
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 charging-profile chgprf1
 virtual-mac      b6:6d:47:47:47:47
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV6 IPV4V6 ]
 upf apn intershat
 dcnr             true
exit
profile dnn intershat1
 network-element-profiles chf chf1
 network-element-profiles amf amf1
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 charging-profile chgprf1
```

```
 virtual-mac      b6:6d:47:47:47:48
 pcscf-profile    PCSCF_Prof_2
 ssc-mode 1
 session type IPV4
exit
profile dnn intershat2
 network-element-profiles chf chf
 network-element-profiles amf amf
 network-element-profiles pcf pcf
 network-element-profiles udm udm
 charging-profile chgprf1
 virtual-mac      b6:6d:47:47:47:49
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV6 IPV4V6 ]
 upf apn intershat2
exit
profile dnn starent.com
 network-element-profiles chf chf1
 network-element-profiles amf amf1
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 charging-profile chgprf1
 virtual-mac      b6:6d:47:47:47:47
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV6 IPV4V6 ]
 upf apn starent.com
exit
profile qos abc
 ambr ul "250 Kbps"
 ambr dl "500 Kbps"
 qi5      7
 arp priority-level 14
 arp preempt-cap NOT_PREEMPT
 arp preempt-vuln PREEMPTABLE
 priority 120
 max data-burst 2000
exit
profile nf-client nf-type udm
 udm-profile UP1
  locality LOC1
   priority 30
   service name type nudm-sdm
    endpoint-profile EP1
     capacity   30
     uri-scheme http
     version
      uri-version v2
      exit
     exit
     endpoint-name EP1
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 8001
     exit
    exit
   exit
   service name type nudm-uecm
    endpoint-profile EP1
     capacity   30
     uri-scheme http
     endpoint-name EP1
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 8001
     exit
    exit
```

```
   exit
   service name type nudm-ee
    endpoint-profile EP1
     capacity       30
     api-uri-prefix PREFIX
     api-root       ROOT
     uri-scheme     http
     endpoint-name EP1
      priority 56
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 8001
     exit
    exit
   exit
  exit
 exit
exit
profile nf-client nf-type pcf
 pcf-profile PP1
  locality LOC1
   priority 30
   service name type npcf-am-policy-control
    endpoint-profile EP1
     capacity   30
     uri-scheme http
     endpoint-name EP1
      priority 56
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 8003
     exit
    exit
   exit
   service name type npcf-smpolicycontrol
    endpoint-profile EP1
     capacity   30
     uri-scheme http
     endpoint-name EP1
      priority 56
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 8003
     exit
    exit
   exit
  exit
 exit
exit
profile nf-client nf-type amf
 amf-profile AP1
  locality LOC1
   priority 30
   service name type namf-comm
    endpoint-profile EP2
     capacity   30
     uri-scheme http
     endpoint-name EP1
      priority 56
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 8002
     exit
    exit
   exit
  exit
 exit
exit
```

```
profile nf-client nf-type chf
 chf-profile CP1
  locality LOC1
   priority 30
   service name type nchf-convergedcharging
    endpoint-profile EP1
     capacity   30
     uri-scheme http
     version
      uri-version v2
      exit
     exit
     endpoint-name EP1
      priority 56
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 8004
     exit
    exit
   exit
  exit
 exit
 chf-profile CP2
  locality LOC1
   priority 31
   service name type nchf-convergedcharging
    endpoint-profile EP1
     capacity   30
     uri-scheme http
     version
      uri-version v2
      exit
     exit
     endpoint-name EP1
      priority 56
      primary ip-address ipv4 209.165.201.21
      primary ip-address port 9040
     exit
    exit
   exit
  exit
 exit
exit
profile nf-pair nf-type UDM
 nrf-discovery-group udmdiscovery
 locality client  LOC1
 locality preferred-server LOC1
 locality geo-server GEO
exit
profile nf-pair nf-type AMF
 nrf-discovery-group udmdiscovery
 locality client  LOC1
 locality preferred-server LOC1
 locality geo-server GEO
exit
profile nf-pair nf-type PCF
 nrf-discovery-group udmdiscovery
 locality client  LOC1
 locality preferred-server LOC1
 locality geo-server GEO
exit
profile nf-pair nf-type UPF
 nrf-discovery-group udmdiscovery
 locality client  LOC1
 locality preferred-server LOC1
```

```
 locality geo-server GEO
exit
profile nf-pair nf-type CHF
 nrf-discovery-group udmdiscovery
 locality client  LOC1
 locality preferred-server LOC1
 locality geo-server GEO
exit
profile nf-client-failure nf-type udm
 profile failure-handling FH4
  service name type nudm-sdm
   message type UdmSdmGetUESMSubscriptionData
    status-code httpv2 403
     retry  3
     action retry-and-ignore
    exit
    status-code httpv2 404
     action continue
    exit
    status-code httpv2 413
     retry  3
     action retry-and-continue
    exit
    status-code httpv2 501,504
     retry  3
     action retry-and-terminate
    exit
    status-code httpv2 503
     action terminate
    exit
   exit
   message type UdmSdmSubscribeToNotification
    status-code httpv2 403
     retry  3
     action retry-and-ignore
    exit
    status-code httpv2 404
     action continue
    exit
    status-code httpv2 413
     retry  3
     action retry-and-continue
    exit
    status-code httpv2 501,504
     retry  3
     action retry-and-terminate
    exit
    status-code httpv2 503
     action terminate
    exit
   exit
  exit
  service name type nudm-uecm
   message type UdmUecmRegisterSMF
    status-code httpv2 403
     retry  3
     action retry-and-ignore
    exit
    status-code httpv2 404
     action continue
    exit
    status-code httpv2 413
     retry  3
     action retry-and-continue
```

```
    exit
    status-code httpv2 501,504
     retry  3
     action retry-and-terminate
    exit
    status-code httpv2 503
     action terminate
    exit
   exit
  exit
 exit
exit
profile nf-client-failure nf-type pcf
 profile failure-handling FH1
  service name type npcf-smpolicycontrol
   message type PcfSmpolicycontrolCreate
    status-code httpv2 0,403
     action retry-and-ignore
    exit
    status-code httpv2 400
     action continue
    exit
    status-code httpv2 404
     action terminate
    exit
    status-code httpv2 500
     retry  2
     action retry-and-ignore
    exit
    status-code httpv2 503
     retry  2
     action retry-and-continue
    exit
   exit
   message type PcfSmpolicycontrolUpdate
    status-code httpv2 0,403
     action retry-and-ignore
    exit
    status-code httpv2 400
     action continue
    exit
    status-code httpv2 404
     action terminate
    exit
    status-code httpv2 500
     retry  2
     action retry-and-ignore
    exit
    status-code httpv2 503
     retry  2
     action retry-and-continue
    exit
   exit
   message type PcfSmpolicycontrolDelete
    status-code httpv2 0,403
     action retry-and-ignore
    exit
    status-code httpv2 400
     action continue
    exit
    status-code httpv2 404
     action terminate
    exit
    status-code httpv2 500
```

```
      retry  2
      action retry-and-ignore
     exit
     status-code httpv2 503
      retry  2
      action retry-and-continue
     exit
    exit
   exit
  exit
 exit
profile nf-client-failure nf-type chf
 profile failure-handling FH2
  service name type nchf-convergedcharging
   message type ChfConvergedchargingCreate
    status-code httpv2 0,500,504
     action continue
    exit
    status-code httpv2 400,404
     retry  3
     action retry-and-terminate
    exit
    status-code httpv2 403
     retry  3
     action retry-and-ignore
    exit
    status-code httpv2 503
     action terminate
    exit
   exit
   message type ChfConvergedchargingUpdate
    status-code httpv2 0,500,504
     action continue
    exit
    status-code httpv2 400,404
     retry  3
     action retry-and-terminate
    exit
    status-code httpv2 403
     retry  3
     action retry-and-ignore
    exit
    status-code httpv2 503
     action terminate
    exit
   exit
   message type ChfConvergedchargingDelete
    status-code httpv2 0,500,504
     action continue
    exit
    status-code httpv2 400,404
     retry  3
     action retry-and-terminate
    exit
    status-code httpv2 403
     retry  3
     action retry-and-ignore
    exit
    status-code httpv2 503
     action terminate
    exit
   exit
  exit
 exit
```

```
exit
profile smf smf1
 locality      LOC1
 allowed-nssai [ slice1 ]
 plmn-id mcc 123
 plmn-id mnc 456
 service name nsmf-pdu
  type               pdu-session
  schema             http
  service-id         1
  version            1.Rn.0.0
  http-endpoint base-url http://smf-service
  icmpv6-profile     icmpprf1
  compliance-profile comp1
  access-profile     access1
  subscriber-policy  polSub
 exit
exit
profile sgw sgw1
 sgw-charging-threshold thre1
 sgw-charging-profile   ch1
 locality               LOC2
 fqdn                   cisco.com.apn.epc.mnc456.mcc123
 charging-mode          gtpp
exit
profile sgw-charging-threshold thre1
 cc profile value 1
  volume total 100000
  buckets  1
  duration 60
 exit
 cc profile value 2
  volume uplink 100000
  volume downlink 100000
  buckets  1
  duration 120
 exit
exit
profile sgw-charging-profile ch1
 gtpp-triggers volume-limit enable
 gtpp-triggers time-limit enable
 gtpp-profile pf1
exit
profile gtpp-profile pf1 gtpp
 local-storage
  file
   rotation
    volume        5
    cdr-count     1000
    time-interval 60
   exit
   name
    prefix               NYPCF508
    format               .%Y-%m-%d%H-%M-%S.%4Q
    max-file-seq-num     4
    start-file-seq-num   1
    recover-file-seq-num false
   exit
   purge-processed-files purge-interval 10
   format custom5
  exit
  push
   encrypted-url
"$8$6vhjkoHt8RL2noFs/ON6ZJavTDzwGS2KUn/YqlBzgkezFmx5SzvnARYzAdVacCSyCirYOvcC\nTFnHpBNim3QY3Q=="
```

```
  exit
 exit
 dictionary custom24
exit
policy subscriber polSub
 precedence 1
  sst              02
  sdt              Abf123
  serving-plmn mcc 123
  serving-plmn mnc 456
  supi-start-range 100000000000001
  supi-stop-range  999999999999999
  gpsi-start-range 1000000000
  gpsi-stop-range  9999999999
  operator-policy  opPol1
 exit
 precedence 511
  operator-policy defOprPol1
 exit
exit
policy operator defOprPol1
 policy dnn         defPolDnn
 policy network-capability nc1
exit
policy operator opPol1
 policy dnn         polDnn
 policy network-capability nc1
exit
policy dnn defPolDnn
 profile default-profile
 dnn dnn2 profile profile2
 dnn intershat profile intershat
 dnn intershat1 profile intershat1
 dnn starent.com profile starent.com
exit
policy dnn polDnn
 profile default-profile
 dnn dnn2 profile profile2
 dnn intershat profile intershat
 dnn intershat1 profile intershat1
 dnn intershat2 profile intershat2
 dnn starent.com profile starent.com
exit
policy network-capability nc1
 nw-support-local-address-tft true
exit
nssai name slice1
 sst 2
 sdt Abf123
 dnn [ dnn1 intershat intershat1 intershat2 ]
exit
ipam
 instance 1
  source local
  address-pool poolv4
   vrf-name ISP
   tags
    dnn starent.com
   exit
   ipv4
    split-size
     per-cache 1024
     per-dp    256
```

```
  exit
  address-range 209.165.202.129 209.165.200.253
 exit
exit
address-pool poolv4DNN2
 vrf-name ISP
 tags
  dnn intershat1
 exit
 ipv4
  split-size
   per-cache 1024
   per-dp    256
  exit
  address-range 209.165.200.241 209.165.200.244
 exit
exit
address-pool poolv4DNN3
 vrf-name ISP
 static
 tags
  dnn intershat2
 exit
 ipv4
  split-size
   per-cache 512
   per-dp    512
  exit
  address-range 209.165.200.247 209.165.200.248
 exit
 ipv6
  prefix-ranges
   split-size
    per-cache 8192
    per-dp    8192
   exit
   prefix-range 2002:db0:: length 48
  exit
 exit
exit
address-pool poolv4vDNN
 vrf-name ISP
 tags
  dnn intershat1
 exit
 ipv4
  split-size
   per-cache 1024
   per-dp    256
  exit
  address-range 209.165.200.245 209.165.200.244
 exit
exit
address-pool poolv6
 vrf-name ISP
 tags
  dnn intershat
 exit
 ipv6
  prefix-ranges
   split-size
    per-cache 8192
    per-dp    1024
   exit
```

```
     prefix-range 2001:db0:: length 48
    exit
   exit
  exit
  address-pool poolv6DNN2
   vrf-name ISP
   tags
    dnn intershat1
   exit
   ipv6
    prefix-ranges
     split-size
      per-cache 8192
      per-dp    1024
     exit
     prefix-range 2001:ef0:: length 48
    exit
   exit
  exit
  address-pool poolv6vDNN
   vrf-name ISP
   tags
    dnn intershat1
   exit
   ipv6
    prefix-ranges
     split-size
      per-cache 8192
      per-dp    1024
     exit
     prefix-range 2001:ab0:: length 48
    exit
   exit
  exit
 exit
exit
cdl system-id          1
cdl enable-geo-replication true
cdl deployment-model   small
cdl zookeeper replica 1
cdl remote-site 2
 db-endpoint host 209.165.202.157
 db-endpoint port 8882
 kafka-server 209.165.202.157 10001
 exit
exit
cdl datastore session
 geo-remote-site [ 2 ]
 slice-names     [ cnSGW1 cnSGW2 ]
 endpoint replica 1
 endpoint external-ip 209.165.202.156
 endpoint external-port 8882
 index map    1
 index write-factor 1
 slot replica 1
 slot map     1
 slot write-factor 1
 features instance-aware-notification enable true
 features instance-aware-notification system-id 1
  slice-names [ cnSGW1 ]
 exit
 features instance-aware-notification system-id 2
  slice-names [ cnSGW2 ]
 exit
```

```
exit
cdl kafka replica 1
cdl kafka external-ip 209.165.202.156 10001
exit
etcd replicas 1
instance instance-id 1
 endpoint li
  replicas 1
  vip-ip 209.165.200.237
 exit
 endpoint nodemgr
  replicas 1
  nodes    1
 exit
 endpoint gtp
  replicas 1
  interface s5
   vip-ip 209.165.201.11
  exit
  interface s5e
   vip-ip 209.165.201.21
  exit
  interface s11
   vip-ip 209.165.200.237
  exit
 exit
 endpoint pfcp
  replicas 1
  interface sxa
   heartbeat
    interval 0
   exit
  exit
  interface n4
   heartbeat
    interval               0
    retransmission-timeout 3
    max-retransmissions    5
   exit
  exit
 exit
 endpoint radius-dns
  replicas 1
  vip-ip 209.165.201.21
 exit
 endpoint service
  replicas 1
 exit
 endpoint protocol
  replicas     1
  internal-vip 209.165.201.11
  vip-ip 209.165.201.21
  interface sxa
   vip-ip 209.165.201.21
  exit
  interface n4
   vip-ip 209.165.201.11
  exit
 exit
 endpoint gtpprime
  replicas 2
  nodes    1
 exit
 endpoint sgw-service
```

```
  replicas 1
 exit
 endpoint geo
  replicas 1
  nodes    2
  interface geo-internal
   vip-ip 209.165.200.233 vip-port 7001
  exit
  interface geo-external
   vip-ip 209.165.200.234 vip-port 7002
  exit
 exit
 endpoint sbi
  replicas 1
  vip-ip 209.165.201.21
 exit
 endpoint bgpspeaker
  replicas 1
  nodes    2
 exit
exit
instance instance-id 2
 endpoint li
  replicas 1
  vip-ip 209.165.200.238
 exit
 endpoint nodemgr
  replicas 1
  nodes    1
 exit
 endpoint gtp
  replicas 1
  interface s5
   vip-ip 209.165.201.12
  exit
  interface s5e
   vip-ip 209.165.201.141
  exit
  interface s11
   vip-ip 209.165.200.238
  exit
 exit
 endpoint pfcp
  replicas 1
  interface sxa
   heartbeat
    interval 0
   exit
  exit
  interface n4
   heartbeat
    interval               0
    retransmission-timeout 3
    max-retransmissions    5
   exit
  exit
 exit
 endpoint radius-dns
  replicas 1
  vip-ip 209.165.201.141
 exit
 endpoint service
  replicas 1
 exit
```

```
 endpoint protocol
  replicas     1
  internal-vip 209.165.201.11
  vip-ip 209.165.201.141
  interface sxa
   vip-ip 209.165.201.141
  exit
  interface n4
   vip-ip 209.165.201.12
  exit
 exit
 endpoint gtpprime
  replicas 2
  nodes    1
 exit
 endpoint sgw-service
  replicas 1
 exit
 endpoint geo
  replicas 1
  nodes    2
  interface geo-internal
   vip-ip 209.165.200.235 vip-port 7001
  exit
  interface geo-external
   vip-ip 209.165.200.236 vip-port 7002
  exit
 exit
 endpoint sbi
  replicas 1
  vip-ip 209.165.201.141
 exit
 endpoint bgpspeaker
  replicas 1
  nodes    2
 exit
exit
logging level application debug
logging level transaction debug
logging level tracing debug
logging name gtpp-ep0.application.config level application debug
logging name gtpp-ep0.application.gen level application trace
logging name gtpp-ep1.application.config level application debug
logging name gtpp-ep1.application.gen level application trace
logging name infra.cdr.core level application debug
logging name infra.cdr_sftp.core level application debug
logging name infra.config.core level application trace
logging name infra.config.core level transaction trace
logging name infra.config.core level tracing off
logging name infra.message_log.core level transaction trace
router bgp 65061
 bfd interval 250000 min_rx 250000 multiplier 3
 interface v4001
  neighbor 209.165.202.131 remote-as 65060 fail-over bfd
 exit
 policy-name allow-all ip-prefix 209.165.201.30/0 mask-range 0..32
exit
deployment
 app-name     smf
 cluster-name Local
 dc-name      DC
 model        small
exit
k8 label protocol-layer key disktype value ssd
```

```
exit
geomonitor podmonitor pods bgpspeaker-pod
 retryCount            1
 retryInterval         200
 retryFailOverInterval 200
 failedReplicaPercent  40
exit
geomonitor podmonitor pods gtpp-ep
 retryCount            1
 retryInterval         200
 retryFailOverInterval 200
 failedReplicaPercent  40
exit
geomonitor podmonitor pods li-ep
 retryCount            1
 retryInterval         200
 retryFailOverInterval 200
 failedReplicaPercent  40
exit
geomonitor podmonitor pods sgw-service
 retryCount            1
 retryInterval         200
 retryFailOverInterval 200
 failedReplicaPercent  40
exit
instances instance 1
 system-id  DCNAME001
 cluster-id CLUSTER0001
 slice-name cnSGW1
exit
instances instance 2
 system-id  DCNAME002
 cluster-id CLUSTER0002
 slice-name cnSGW2
exit
local-instance instance 1
system mode running
helm default-repository cn
helm repository cn
 access-token
sgw-deployer.gen:AKCp8ihVrCfvm9puwTSt8oKGK6HxP1Fn8sLY5fzqWyR3NhrBmjjJrUHaxfZD3ziQpiLkAy1Q3

 url
https://engci-maven-master.cisco.com/artifactory/smi-fuse-internal-snapshot/mobile-cnat-cn/cn-products/rel-2021.02/
exit
k8s name           cn
k8s namespace      cn
k8s nf-name        smf
k8s registry       dockerhub.cisco.com/smi-fuse-docker-internal
k8s single-node    true
k8s use-volume-claims true
k8s image-pull-secrets regcred
k8s ingress-host-name 209.165.200.235.nip.io
aaa authentication users user admin
 uid        117
 gid        117
 password   $1$g8J36yTY$1g/tM5a9pdsGMnKcspnxD.
 ssh_keydir /tmp/admin/.ssh
 homedir    /tmp/admin
exit
aaa ios level 0
 prompt "\h> "
exit
aaa ios level 15
```

```
 prompt "\h# "
exit
aaa ios privilege exec
 level 0
  command action
  exit
  command autowizard
  exit
  command enable
  exit
  command exit
  exit
  command help
  exit
  command startup
  exit
 exit
 level 15
  command configure
  exit
 exit
exit
nacm write-default deny
nacm groups group LI
 user-name [ liadmin ]
exit
nacm groups group LI2
 user-name [ liadmin2 ]
exit
nacm groups group LI3
 user-name [ liadmin3 ]
exit
nacm groups group admin
 user-name [ admin ]
exit
nacm rule-list admin
 group [ admin ]
 rule li-deny-tap
  module-name       lawful-intercept
  path              /lawful-intercept
  access-operations *
  action            deny
 exit
 rule li-deny-clear
  module-name       tailf-mobile-smf
  path              /clear/lawful-intercept
  access-operations *
  action            deny
 exit
 rule any-access
  action permit
 exit
exit
nacm rule-list confd-api-manager
 group [ confd-api-manager ]
 rule any-access
  action permit
 exit
exit
nacm rule-list ops-center-security
 group [ * ]
 rule change-self-password
  module-name       ops-center-security
  path              /smiuser/change-self-password
```

```
  access-operations exec
  action            permit
 exit
 rule smiuser
  module-name       ops-center-security
  path              /smiuser
  access-operations exec
  action            deny
 exit
exit
nacm rule-list lawful-intercept
 group [ LI LI2 LI3 ]
 rule li-accept-tap
  module-name       lawful-intercept
  path              /lawful-intercept
  access-operations *
  action            permit
 exit
 rule li-accept-clear
  module-name       tailf-mobile-smf
  path              /clear/lawful-intercept
  access-operations *
  action            permit
 exit
exit
nacm rule-list any-group
 group [ * ]
 rule li-deny-tap
  module-name       lawful-intercept
  path              /lawful-intercept
  access-operations *
  action            deny
 exit
 rule li-deny-clear
  module-name       tailf-mobile-smf
  path              /clear/lawful-intercept
  access-operations *
  action            deny
 exit
exit
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。