Cisco ASR 5x00 Enhanced Charging Services Administration Guide

Version 15.0

Last Updated October 31, 2014

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
     800 553-NETS (6387)
Fax: 408 527-0883
THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

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

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

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

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

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. 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.

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.

Cisco ASR 5x00 Enhanced Charging Services Administration Guide

© 2014 Cisco Systems, Inc. All rights reserved.
# CONTENTS

## Enhanced Charging Service Overview

- Introduction ................................................................................................. 12
- Qualified Platforms .................................................................................. 12
- License Requirements ............................................................................... 12
- Basic Features and Functionality ............................................................. 13
  - Shallow Packet Inspection ................................................................. 13
  - Deep Packet Inspection ...................................................................... 13
  - Charging Subsystem ............................................................................. 13
    - Traffic Analyzers ............................................................................. 13
    - How ECS Works .............................................................................. 14
- ECS Deployment and Architecture ............................................................ 28
- Enhanced Features and Functionality ...................................................... 30
  - Session Control in ECS ...................................................................... 30
  - Service Group QoS Feature ................................................................ 30
    - Configuration Overview ................................................................. 32
    - Support for Service-based QoS ....................................................... 33
    - Hierarchical Enforcement of QoS Parameters ............................... 34
    - Reporting Statistics and Usage to PCRF ......................................... 35
  - Time and Flow-based Bearer Charging in ECS ................................... 35
  - Content Filtering Support .................................................................... 36
    - Content Filtering Server Group Support ......................................... 37
    - In-line Content Filtering Support ................................................... 37
  - DNS Snooping ....................................................................................... 37
    - License Requirements ..................................................................... 38
    - Bulkstatistics Support ...................................................................... 38
    - How it Works ..................................................................................... 38
    - Limitations and Dependencies .......................................................... 42
  - IP Readdressing ..................................................................................... 42
  - URL-based Re-addressing ..................................................................... 43
  - Next-hop Address Configuration .......................................................... 44
  - Post Processing ...................................................................................... 45
    - How the Post-processing Feature Works ......................................... 45
  - Tethering Detection .............................................................................. 46
    - License Requirements ..................................................................... 46
    - MUR Support for Tethering Detection ............................................. 46
    - Tethering Detection Databases ......................................................... 47
    - Loading and Upgrading Tethering Detection Databases .................. 47
  - Session Recovery Support ...................................................................... 47
  - Time-of-Day Activation/Deactivation of Rules ...................................... 47
    - How the Time-of-Day Activation/Deactivation of Rules Feature Works 48
  - Transactional Rule Matching .................................................................. 48
Contents

Fastpath ................................................................. 48
Bulkstatistics Support ........................................... 51
Limitations and Dependencies .................................. 52
Support for WebSocket Protocol Identification ............. 52
How it Works ......................................................... 52
URL Filtering ......................................................... 52
Implementation of AES Encryption ............................ 53
TCP Proxy .............................................................. 54
Flow Admission Control .......................................... 55
TCP Proxy Behavior and Limitations ......................... 55
Dynamic Disabling of TCP Proxy ............................... 58
X-Header Insertion and Encryption .............................. 59
License Requirements .............................................. 59
X-Header Insertion .................................................. 59
X-Header Encryption ............................................... 60
Limitations to the Header Insertion and Encryption Features .................................................. 61
Supported X-Headers ................................................. 62
X-Header Enrichment Anti Spoofing ........................... 63
RAN Bandwidth Optimization ..................................... 64
Selective TFT Suppression for Default Bearer ............... 64
Accounting and Charging Interfaces .......................... 65
GTPP Accounting .................................................... 65
RADIUS Accounting and Credit Control ..................... 65
Diameter Accounting and Credit Control ..................... 66
Gx Interface Support .................................................. 66
Gy Interface Support .................................................. 66
Event Detail Records (EDRs) ....................................... 67
Usage Detail Records (UDRs) ....................................... 69
Charging Methods and Interfaces ............................... 69
Prepaid Credit Control ............................................. 69
Postpaid ................................................................. 69
Prepaid Billing in ECS .............................................. 70
How ECS Prepaid Billing Works ................................. 70
Credit Control Application (CCA) in ECS ................. 71
How Credit Control Application (CCA) Works for Prepaid Billing ............................................... 71
Postpaid Billing in ECS ............................................. 72
How ECS Postpaid Billing Works .............................. 73
External Storage System .......................................... 76
System Resource Allocation ..................................... 77
Redundancy Support in ECS ....................................... 78
Intra-chassis Session Recovery Interoperability ............ 78
Recovery from Task Failure ....................................... 78
Recovery from CPU or Packet Processing Card Failure .... 78
Inter-chassis Session Recovery Interoperability ............ 78
Inter-chassis Session Recovery Architecture .................. 79
Session Recovery Improvements ............................... 79
Impact on xDR File Naming ....................................... 79
Impact on xDR File Content ....................................... 80

Enhanced Charging Service Configuration ........................ 83
Initial Configuration ................................................ 84
Creating the ECS Administrative User Account .............. 84
Installing the ECS License ........................................ 85
Enabling Enhanced Charging Service ......................... 85
Configuring the Enhanced Charging Service .................. 86
Verifying your Configuration..............................................................118
Configuring URL-based Re-addressing ........................................118
Configuring X-Header Insertion and Encryption Feature ...............118
  Configuring X-Header Insertion..................................................118
  Configuring X-Header Encryption .............................................120
About this Guide

This document pertains to the features and functionality that run on and/or that are related to the Cisco® ASR 5000 and virtualized platforms.
## Conventions Used

The following tables describe the conventions used throughout this documentation.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Notice Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Information Note" /></td>
<td>Information Note</td>
<td>Provides information about important features or instructions.</td>
</tr>
<tr>
<td><img src="image" alt="Caution" /></td>
<td>Caution</td>
<td>Alerts you of potential damage to a program, device, or system.</td>
</tr>
<tr>
<td><img src="image" alt="Warning" /></td>
<td>Warning</td>
<td>Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.</td>
</tr>
</tbody>
</table>

### Typeface Conventions

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

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.
Additional Information

Refer to the following guides for supplemental information about the system:

- Cisco ASR 5000 Installation Guide
- Cisco ASR 5000 System Administration Guide
- Cisco ASR 5x00 Command Line Interface Reference
- Cisco ASR 5x00 Thresholding Configuration Guide
- Cisco ASR 5x00 SNMP MIB Reference
- StarOS IP Security (IPSec) Reference
- Web Element Manager Installation and Administration Guide
- Cisco ASR 5x00 AAA Interface Administration and Reference
- Cisco ASR 5x00 GTPP Interface Administration and Reference
- Cisco ASR 5x00 Release Change Reference
- Cisco ASR 5x00 Statistics and Counters Reference
- Cisco ASR 5x00 Gateway GPRS Support Node Administration Guide
- Cisco ASR 5x00 HRPD Serving Gateway Administration Guide
- Cisco ASR 5000 IP Services Gateway Administration Guide
- Cisco ASR 5x00 Mobility Management Entity Administration Guide
- Cisco ASR 5x00 Packet Data Network Gateway Administration Guide
- Cisco ASR 5x00 Packet Data Serving Node Administration Guide
- Cisco ASR 5x00 System Architecture Evolution Gateway Administration Guide
- Cisco ASR 5x00 Serving GPRS Support Node Administration Guide
- Cisco ASR 5x00 Serving Gateway Administration Guide
- Cisco ASR 5000 Session Control Manager Administration Guide
- Cisco ASR 5000 Packet Data Gateway/Tunnel Termination Gateway Administration Guide
- Release notes that accompany updates and upgrades to the StarOS for your service and platform
This chapter provides an overview of the Enhanced Charging Service (ECS) in-line service, also known as Active Charging Service (ACS).

This chapter covers the following topics:

- Introduction
- Basic Features and Functionality
- ECS Deployment and Architecture
- Enhanced Features and Functionality
Introduction

The Enhanced Charging Service (ECS) is an in-line service feature that enables operators to reduce billing-related costs and gives the ability to offer tiered, detailed, and itemized billing to their subscribers. Using shallow and deep packet inspection (DPI), ECS allows operators to charge subscribers based on actual usage, number of bytes, premium services, location, and so on. ECS also generates charging records for postpaid and prepaid billing systems.

The ECS is an enhanced or extended premium service. The System Administration Guide provides basic system configuration information, and the product administration guides provide information to configure the core network service functionality. It is recommended that you select the configuration example that best meets your service model, and configure the required elements for that model before using the procedures in this document.

Qualified Platforms

ECS is a StarOS in-line service application that runs on Cisco ASR 5x00 platforms. For additional platform information, refer to the appropriate System Administration Guide and/or contact your Cisco account representative.

License Requirements

The ECS in-line service is a licensed Cisco feature. Separate session and feature licenses may be required. Contact your Cisco account representative for detailed information on specific licensing requirements.

For information on installing and verifying licenses, refer to the Managing License Keys section of the Software Management Operations chapter in the System Administration Guide.
Basic Features and Functionality

This section describes basic features of the ECS in-line service.

Shallow Packet Inspection

Shallow packet inspection is the examination of the layer 3 (IP header) and layer 4 (for example, UDP or TCP header) information in the user plane packet flow. Shallow packet analyzers typically determine the destination IP address or port number of a terminating proxy.

Deep Packet Inspection

Deep-packet inspection is the examination of layer 7, which contains Uniform Resource Identifier (URI) information. In some cases, layer 3 and 4 analyzers that identify a trigger condition are insufficient for billing purposes, so layer 7 examination is used. Whereas, deep-packet analyzers typically identify the destination of a terminating proxy.

For example, if the Web site “www.companyname.com” corresponds to the IP address 1.1.1.1, and the stock quote page (www.companyname.com/quotes) and the company page (www.companyname.com/business) are chargeable services, while all other pages on this site are free. Because all parts of this Web site correspond to the destination address of 1.1.1.1 and port number 80 (http), determination of chargeable user traffic is possible only through the actual URL (layer 7).

DPI performs packet inspection beyond layer 4 inspection and is typically deployed for:

- Detection of URI information at level 7 (for example, HTTP, WTP, RTSP URLs)
- Identification of true destination in the case of terminating proxies, where shallow packet inspection would only reveal the destination IP address/port number of a terminating proxy such as the OpCo’s WAP gateway
- De-encapsulation of nested traffic encapsulation, for example MMS-over-WTP/WSP-over-UDP/IP
- Verification that traffic actually conforms to the protocol the layer 4 port number suggests

Charging Subsystem

ECS has protocol analyzers that examine uplink and downlink traffic. Incoming traffic goes into a protocol analyzer for packet inspection. Routing rules definitions (ruledefs) are applied to determine which packets to inspect. This traffic is then sent to the charging engine where charging rules definitions are applied to perform actions such as block, redirect, or transmit. These analyzers also generate usage records for the billing system.

Traffic Analyzers

Traffic analyzers in ECS are based on configured ruledefs. Ruledefs used for traffic analysis analyze packet flows and create usage records. The usage records are created per content type and forwarded to a prepaid server or to a billing system.

The Traffic Analyzer function can perform shallow (layer 3 and layer 4) and deep (above layer 4) packet inspection of IP packet flows. It is able to correlate all layer 3 packets (and bytes) with higher layer trigger criteria (for example, URL detected in an HTTP header). It also performs stateful packet inspection for complex protocols like FTP, RTSP, and SIP that dynamically open ports for the data path and this way, user plane payload is differentiated into “categories”. Traffic
Enhanced Charging Service Overview

Basic Features and Functionality

Traffic analyzers can also detect video streaming over RTSP, and image downloads and MMS over HTTP and differential treatment can be given to the Vcast traffic.

Traffic analyzers work at the application level as well, and perform event-based charging without the interference of the service platforms.

The ECS content analyzers can inspect and maintain state across various protocols at all layers of the OSI stack. The ECS supports inspecting and analyzing the following protocols:

- Domain Name System (DNS)
- File Transfer Protocol (FTP)
- Hyper Text Transfer Protocol (HTTP)
- Internet Control Message Protocol (ICMP)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Message Access Protocol (IMAP)
- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Multimedia Messaging Service (MMS)
- Post Office Protocol version 3 (POP3)
- Remote Authentication Dial In User Service (RADIUS)
- RTP Control Protocol/Real-time Transport Control Protocol (RTCP)
- Real-time Transport Protocol (RTP)
- Real Time Streaming Protocol (RTSP)
- Session Description Protocol (SDP)
- Secure-HTTP (S-HTTP)
- Session Initiation Protocol (SIP)
- Simple Mail Transfer Protocol (SMTP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- WebSocket Protocol
- Wireless Session Protocol (WSP)
- Wireless Transaction Protocol (WTP)

Apart from the above protocols, ECS also supports analysis of downloaded file characteristics (for example, file size, chunks transferred, and so on) from file transfer protocols such as HTTP and FTP.

How ECS Works

This section describes the base components of the ECS solution, and the roles they play.

Content Service Steering

Content Service Steering (CSS) enables directing selective subscriber traffic into the ECS subsystem (in-line services internal to the system) based on the content of the data presented by mobile subscribers.
CSS uses Access Control Lists (ACLs) to redirect selective subscriber traffic flows. ACLs control the flow of packets into and out of the system. ACLs consist of “rules” (ACL rules) or filters that control the action taken on packets matching the filter criteria.

ACLs are configurable on a per-context basis and applies to a subscriber through either a subscriber profile (for PDSN) or an APN profile (for GGSN) in the destination context.

**Important:** For more information on CSS, refer to the *Content Service Steering* chapter of the *System Administration Guide*. For more information on ACLs, refer to the *IP Access Control Lists* chapter of the *System Administration Guide*.

---

**Protocol Analyzer**

The Protocol Analyzer is the software stack responsible for analyzing the individual protocol fields and states during packet inspection.

The Protocol Analyzer performs two types of packet inspection:

- **Shallow Packet Inspection**—Inspection of the layer 3 (IP header) and layer 4 (for example, UDP or TCP header) information.
- **Deep Packet Inspection**—Inspection of layer 7 and 7+ information. DPI functionality includes:
  - Detection of Uniform Resource Identifier (URI) information at level 7 (for example, HTTP, WTP, and RTSP URLs)
  - Identification of true destination in the case of terminating proxies, where shallow packet inspection would only reveal the destination IP address/port number of a terminating proxy
  - De-encapsulation of upper layer protocol headers, such as MMS-over-WTP, WSP-over-UDP, and IP-over GPRS
  - Verification that traffic actually conforms to the protocol the layer 4 port number suggests

The Protocol Analyzer performs a stateful packet inspection of complex protocols, such as FTP, RTSP, and SIP, which dynamically open ports for the data path, so the payload can be classified according to content.

The Protocol Analyzer is also capable of determining which layer 3 packets belong (either directly or indirectly) to a trigger condition (for example, URL). In cases where the trigger condition cannot be uniquely defined at layers 3 and 4, then the trigger condition must be defined at layer 7 (that is, a specific URL must be matched).

---

**Protocol Analyzer Software Stack**

Every packet that enters the ECS subsystem must first go through the Protocol Analyzer software stack, which comprises of individual protocol analyzers for each of the supported protocols.
Figure 1. ECS Protocol Analyzer Stack

Note that protocol names are used to represent the individual protocol analyzers.

Each analyzer consists of fields and states that are compared to the protocol-fields and protocol-states in the incoming packets to determine packet content.

**Important:** In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

**Rule Definitions**

Rule definitions (ruledefs) are user-defined expressions based on protocol fields and protocol states, which define what actions to take on packets when specified field values match.
Rule expressions may contain a number of operator types (string, =, >, and so on) based on the data type of the operand. For example, “string” type expressions like URLs and host names can be used with comparison operators like “contains”, “!contains”, “=”, “! =”, “starts-with”, “ends-with”, “!starts-with” and “!ends-with”. In 14.0 and later releases, ECS also supports regular expression based rule matching. For more information, refer to the Regular Expression Support for Rule Matching section.

Integer type expressions like “packet size” and “sequence number” can be used with comparison operators like “=”, “!=”, “>”, “<” or “<=”. Each ruledef configuration consists of multiple expressions applicable to any of the fields or states supported by the respective analyzers.

Ruledefs are of the following types:

- **Routing Ruledefs**—Routing ruledefs are used to route packets to content analyzers. Routing ruledefs determine which content analyzer to route the packet to when the protocol fields and/or protocol-states in ruledef expression are true. Up to 256 ruledefs can be configured for routing.

- **Charging Ruledefs**—Charging ruledefs are used to specify what action to take based on the analysis done by the content analyzers. Actions can include redirection, charge value, and billing record emission. Up to 2048 charging ruledefs can be configured in the system.

- **Post-processing Ruledefs**—Used for post-processing purposes. Enables processing of packets even if the rule matching for them has been disabled.

### Important: When a ruledef is created, if the rule-application is not specified for the ruledef, by default the system considers the ruledef as a charging ruledef.

Ruledefs support a priority configuration to specify the order in which the ruledefs are examined and applied to packets. The names of the ruledefs must be unique across the service or globally. A ruledef can be used across multiple rulebases.

### Important: Ruledef priorities control the flow of the packets through the analyzers and control the order in which the charging actions are applied. The ruledef with the lowest priority number invokes first. For routing ruledefs, it is important that lower level analyzers (such as the TCP analyzer) be invoked prior to the related analyzers in the next level (such as HTTP analyzer and S-HTTP analyzers), as the next level of analyzers may require access to resources or information from the lower level. Priorities are also important for charging ruledefs as the action defined in the first matched charging rule apply to the packet and ECS subsystem disregards the rest of the charging ruledefs.

Each ruledef can be used across multiple rulebases, and up to 2048 ruledefs can be defined in a charging service.

In 14.1 and earlier releases, a maximum of 10 rule expressions (rule-lines) can be added in one ruledef.

In 15.0 and later releases, a maximum of 32 rule expressions (rule-lines) can be added in one ruledef.

Ruledefs have an expression part, which matches specific packets based upon analyzer field variables. This is a boolean (analyzer_field operator value) expression that tests for analyzer field values.

The following is an example of a ruledef to match packets:

```plaintext
http url contains cnn.com
```

-or-

```plaintext
http any-match = TRUE
```

In the following example the ruledef named “rule-for-http” routes packets to the HTTP analyzer:

```plaintext
route priority 50 ruledef rule-for-http analyzer http
```
Where, `rule-for-http` has been defined with the expressions: `tcp either-port = 80`
The following example applies actions where:

- Subscribers whose packets contain the expression “bbc-news” are not charged for the service.
- All other subscribers are charged according to the duration of use of the service.

```plaintext
ruledef port-80
  tcp either-port = 80
  rule-application routing
  exit
ruledef bbc-news
  http url starts-with http://news.bbc.co.uk
  rule-application charging
  exit
ruledef catch-all
  ip any-match = TRUE
  rule-application charging
  exit
charging-action free-site
  content-id 100
  [ ... ]
  exit
charging-action charge-by-duration
  content-id 101
  [ ... ]
  exit
rulebase standard
  [ ... ]
route priority 1 ruledef port-80 analyzer http
  action priority 101 ruledef bbc-news charging-action free-site
  action priority 1000 ruledef catch-all charging-action charge-by-duration
```
The following figure illustrates how ruledefs interact with the Protocol Analyzer Stack and Action Engine to produce charging records.

**Figure 2. ECS In-line Service Processing**

Packets entering the ECS subsystem must first pass through the Protocol Analyzer Stack where routing ruledefs apply to determine which packets to inspect. Then output from this inspection is passed to the charging engine, where charging ruledefs apply to perform actions on the output.

**Routing Ruledefs and Packet Inspection**

The following figure and the steps describe the details of routing ruledef application during packet inspection.
Step 1  The packet is redirected to ECS based on the ACLs in the subscriber’s template /APN and packets enter ECS through the Protocol Analyzer Stack.

Step 2  Packets entering Protocol Analyzer Stack first go through a shallow inspection by passing through the following analyzers in the listed order:

   Step a  Bearer Analyzer
   Step b  IP Analyzer
   Step c  ICMP, TCP, or UDP Analyzer as appropriate

Important: In the current release traffic routes to the ICMP, TCP, and UDP analyzers by default. Therefore, defining routing ruledefs for these analyzers is not required.

Step 3  The fields and states found in the shallow inspection are compared to the fields and states defined in the routing ruledefs in the subscriber’s rulebase.

   The ruledefs’ priority determines the order in which the ruledefs are compared against packets.

Step 4  When the protocol fields and states found during the shallow inspection match those defined in a routing ruledef, the packet is routed to the appropriate layer 7 or 7+ analyzer for deep-packet inspection.

Step 5  After the packet has been inspected and analyzed by the Protocol Analyzer Stack:

   Step a  The packet resumes normal flow and through the rest of the ECS subsystem.
Step b  The output of that analysis flows into the charging engine, where an action can be applied. Applied actions include redirection, charge value, and billing record emission.

Charging Ruledefs and the Charging Engine

This section describes details of how charging ruledefs are applied to the output from the Protocol Analyzer Stack. The following figure and the steps that follow describe the process of charging ruledefs and charging engines.

**Figure 4.  Charging Ruledefs and Charging Engine**

Step 1  In the Classification Engine, the output from the deep-packet inspection is compared to the charging ruledefs. The priority configured in each charging ruledef specifies the order in which the ruledefs are compared against the packet inspection output.
Step 2 When a field or state from the output of the deep-packet inspection matches a field or state defined in a charging ruledef, the ruledef action is applied to the packet. Actions can include redirection, charge value, or billing record emission. It is also possible that a match does not occur and no action will be applied to the packet at all.

Regular Expression Support for Rule Matching

This section describes ECS support for regular expression (regex) rule matching.

In this release, ECS supports regex rule matching only for the following string-based rules:

- http host
- http referer
- http uri
- http url
- rtsp uri
- wsp url
- www url

The following table lists the special characters that you can use in regex rule expressions.

<table>
<thead>
<tr>
<th>Regex Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Zero or more characters</td>
</tr>
<tr>
<td>+</td>
<td>Zero or more repeated instances of the token preceding the +</td>
</tr>
<tr>
<td>?</td>
<td>Match zero or one character</td>
</tr>
<tr>
<td>?</td>
<td>Match the question mark (&lt;ctrl-v&gt;??) character</td>
</tr>
<tr>
<td>+</td>
<td>Match the plus character</td>
</tr>
<tr>
<td>*</td>
<td>Match the asterisk character</td>
</tr>
<tr>
<td>\a</td>
<td>Match the alert (ASCII 7) character</td>
</tr>
<tr>
<td>\b</td>
<td>Match the backspace (ASCII 8) character</td>
</tr>
<tr>
<td>\f</td>
<td>Match the form-feed (ASCII 12) character</td>
</tr>
<tr>
<td>\n</td>
<td>Match the new line (ASCII 10) character</td>
</tr>
</tbody>
</table>

Important: The CLI does not support configuring “?” directly, you must instead use “\077”. For example, if you want to match the string “xyz<any one character>pqr”, you must configure it as: http host regex “xyz\077pqr” In another example, if you want to exactly match the string “url?resource=abc”, you must configure it as: http uri regex “url\\\077resource=abc” Where, the first “\” (backslash) is for the escaping of “?” and then “\077” for specifying “?” to the CLI.
### Regex Character Description

<table>
<thead>
<tr>
<th>Regex Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\r</td>
<td>Match the carriage return (ASCII 13) character</td>
</tr>
<tr>
<td>\t</td>
<td>Match the tab (ASCII 9) character</td>
</tr>
<tr>
<td>\v</td>
<td>Match the vertical tab (ASCII 11) character</td>
</tr>
<tr>
<td>\0</td>
<td>Match the null (ASCII 0) character</td>
</tr>
<tr>
<td>\</td>
<td>Match the backslash character</td>
</tr>
<tr>
<td>Bracketed range [0-9]</td>
<td>Match any single character from the range</td>
</tr>
<tr>
<td>A leading ^ in a range</td>
<td>Do not match any in the range. All other characters represent themselves.</td>
</tr>
<tr>
<td>.x##</td>
<td>Any ASCII character as specified in two-digit hex notation. For example, \x5A yields a “Z”.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Specify OR regular expression operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Important:</td>
<td>When using the regex operator “</td>
</tr>
</tbody>
</table>

The following are some examples of the use of regex characters in rule expressions:

- The following command specifies a regex rule expression using the regex character * (asterisk) to match any of the following or similar values in the HTTP Host request-header field: host1, host101, host23w01.

  ```
  http host regex "host*1"
  ```

- The following command specifies a regex rule expression using the regex character + (plus) to match any of the following or similar values in the HTTP Host request-header field: host1, host101, host23w01.

  ```
  http host regex "host+"
  ```

- The following command specifies a regex rule expression using the regex character \077 (?) to match any of the following or similar values in the HTTP Host request-header field: host101.

  ```
  http host regex "host\077t101"
  ```

- The following command specifies a regex rule expression using the regex character \ (escaped character) to match the following value in the HTTP Host request-header field: host?example.

  ```
  http host regex "host\077example"
  ```

The first two \ form an escape sequence and \077 is converted to ?. The \? is converted to ? as a character and not a place-holder.

- The following command specifies a regex rule expression using the regex character \ (escaped backslash character) to match the following value in the HTTP Host request-header field: host*01.

  ```
  http host regex "host\*01"
  ```

The first \ is used as an escape sequence for the second \. 
- The following command specifies a regex rule expression using the regex character `\+` (escaped `+` character) to match the following value in the HTTP Host request-header field: `host\+01`.

  `http host regex "host\+01"`

- The following command specifies a regex rule expression using the regex character `\` (escaped backslash character) to match the following value in the HTTP Host request-header field: `host\01`.

  `http host regex "host\01"`

- The following command specifies regex rule expression using the regex `[0-9]` to match any of the following or similar values in the HTTP Host request-header field: `hostaBc, hostXyZ, hosthost`. Values starting with the word "host" and not containing numbers.

  `http host regex "host[^0-9]"`

- The following command specifies regex rule expression using the regex `[a-z]` to match any of the following or similar values in the HTTP Host request-header field: `hostabc, hostxyz, hosthost`. Values starting with the word "host" and containing only lowercase letters.

  `http host regex "host[a-z]"`

- The following command specifies a regex rule expression using the regex `|` (or) to match either of the following values in the HTTP Host request-header field: `host1, host23w01`.

  `http host regex "host1|host23w01"`

- The following command defines a regex rule expression to match any of the following or similar values in the RTSP URI string: `rtsp://pvs29p.cvf.fr:554/t1/live/Oui17, rtsp://pvs00p.cvf.fr:554/t1/live/Nrj12, rtsp://pvs90p.cvf.fr:554/t1/live/France24_fr`.

  `rtsp uri regex "rtsp://pvs([0-9][0-9])p.cvf.fr:554/t1/live/(Gulli|Tf1|Tmc|Nrj12|France24_fr|Oui17)*"`


  `www url regex "http://(tp2.site.com|134.210.11.3)/httppvc_clnsssite.com.wap.symphonieserver.musicwaver.com/"`

- The following command defines a regex rule expression to match any of the following or similar values in the WSP URL string: `wsp://home.opera.yahoo.com, wsp://dwld.yahoo.com, wsp://dwld2.yahoo.com`.

  `wsp url regex "wsp://(dwld|opera|home.opera|dwld[1-3]).yahoo.com"`

- The following command defines a regex rule expression to match any of the following or similar values in the HTTP URL string: `http://yahoo.com, http://www.yahoo.co.in, http://yahoo.com/news`.

  `http url regex "(http://|http://www).yahoo.(co.in|com)"`


### How it Works

This section describes how regex rule matching works.

The following steps describe how regex rule matching works:

1. Regex rules/def/group-of-rules/dfs are configured in the CLI.
Regexpruledefsare ruledefs that contain regexpruleexpressions. A ruledef can contain both regexprule and reguleruleexpressions.

Regexprroup-of-ruledefsare group-of-ruledefs that contain regexpruledefs. A group-of-ruledefs can contain both regexprule and reguleruleruledefs.

2. After the regexpruledefs are configured, on the expiry of an internal 30secondtimer, building of the regexpruleengines is triggered.

Note that one regexpruleengine is built per each regexpruleexpressiontype.

Just as with first-time or incremental configurations, SessCtrl/SessMgr recovery/reconciliation also triggers the building of regexpruleengines.

3. The regexpruleengine matches the regexprulestring (specified in the regexpruleexpression) against live traffic, and returns all matching ruledefs.

4. The rule matches are then verified with those configured in the rulebase to determine the best matching rule.

Limitations and Dependencies

This section lists known limitations and restrictions to regexprulematching.

- Changes to ruledefs cause the optimization engines to get updated, hence any changes to ruledefs must be done with care. Preferably during low load times.
- Addition, modification, and deletion of regexpruledefswill result in rebuilding of regexpruleengines, which is time consuming and resource intensive. While the engines are being rebuilt, rule-matching based on the old engines and old configurations may yield inconsistent results.
- Addition, modification, and deletion of action priority lines inside the rulebase has no impact on the regexpruleengines. The regexpruleengines remain intact and the removed action priorities from the rulebase are ignored during rule matching. Similarly, addition, modification (adding or removing ruledefs from it), or deletion of a group-of-ruledefshas no impact on regexpruleengines.
- When adding regexpruledefus, use the following guidelines:
  - Avoid configuring “xyz*” or “*xyz*” type of regexpruleexpression as they are known to consume large memory. Instead, configure Aho-Corasick rules using "starts-with xyz" or "contains xyz" constructs, which comparatively consume less memory. The “starts-with” and “contains” operators are specially tailored for these types of operations, and work much faster (with lot less memory) than the corresponding “regexprule xyz*” or “regexprule *xyz*” operators. Hence, it is recommended that the “starts-with” and “contains” approach be preferred.
  - Do not configure rules frequently. Push as much configuration as possible simultaneously so that all the regexprule rules are available for engine building at the same time. Frequent configuration changes may result in infinite loops with wasted memory and CPU cycles.
  - Do not configure large number of regexprule rules as memory utilization will be high depending on the type of regexprule rules.
  - Frequently monitor status of the engine using the show active-charging regexprule status { all | instance <instance> } CLI command in the Exec Mode. Where <instance> is the SessMgr instance number.
- When deleting ruledefs use the following guidelines:
  - Avoid deleting ruledefs at heavy loads, instead remove them from the required rulebases using the no action priority <action_priority> CLI command in the ACS Rulebase Configuration Mode. Doing so has no impact on regexprule building, although it uses additional memory there is no impact on traffic processing.
Deletion of ruledefs must be done during low load times. As described earlier, it is highly recommended that ruledefs be added, modified, or deleted in bulk, as it results in optimization engine updates.

**Group of Ruledefs**

Group-of-Ruledefs enable grouping ruledefs into categories. When a group-of-ruledefs is configured in a rulebase and any of the ruledefs within the group matches, the specified charging-action is applied and action instances are not processed further.

A group-of-ruledefs may contain optimizable ruledefs. Whether a group is optimized or not is decided on whether all the ruledefs in the group-of-ruledefs can be optimized, and if the group is included in a rulebase that has optimization turned on.

When a new ruledef is added, it is checked if it is included in any group-of-ruledefs, and whether it requires optimization.

The group-of-ruledefs configuration enables setting the application for the group (group-of-ruledefs-application parameter). When set to gx-alias, the group-of-ruledefs is expanded only to extract the rule names out of it (with their original priority and charging actions) ignoring the field priority set within the group. This is just an optimization over the PCRF to PCEF interface where a need to install/remove a large set of predefined rules at the same time exists. Though this is possible over the Gx interface (with a limit of 256), it requires a large amount of PCRF resources to encode each name. This also increases the message size.

This aliasing function enables to group a set of ruledef names and provides a simple one-name alias that when passed over Gx, as a Charging-Rule-Base-Name AVP, is expanded to the list of names with each rule being handled individually. From the PCEF point of view, it is transparent, as if the PCRF had activated (or deactivated) those rules by naming each one.

In 14.1 and earlier releases, a maximum of 128 ruledefs can be added to a group-of-ruledefs, and a maximum of 64 group-of-ruledefs can be configured.

In 15.0 and later releases, a maximum of 128 ruledefs can be added to a group-of-ruledefs, and a maximum of 128 group-of-ruledefs can be configured.

**Rulebase**

A rulebase allows grouping one or more rule definitions together to define the billing policies for individual subscribers or groups of subscribers.

A rulebase is a collection of ruledefs and their associated billing policy. The rulebase determines the action to be taken when a rule is matched. A maximum of 512 rulebases can be specified in the ECS service.

It is possible to define a ruledef with different actions. For example, a Web site might be free for postpaid users and charge based on volume for prepaid users. Rulebases can also be used to apply the same ruledefs for several subscribers, which eliminate the need to have unique ruledefs for each subscriber.

**Rulebase List**

A rulebase list allows grouping one or more rulebases together, enabling the Online Charging System (OCS) to choose the rulebase for a subscriber from the rulebase list.

A rulebase list enables a list of rulebases to be sent to the OCS over the Gy interface using a buffer. The OCS can then select a specific rulebase from the rulebase list, and apply the ruledefs and billing policies associated with that rulebase to subscribers.
Rulebase lists are created and configured in the ACS Configuration Mode. The maximum length of an individual rulebase-list name is 64 bytes. The buffer that stores space-separated rulebase names within a rulebase-list is of 256 bytes.

In 12.3 and earlier releases, a maximum of 20 rulebase lists can be configured per active charging service.
In 14.0 and later releases, a maximum of 128 rulebase lists can be configured per active charging service.

When a subscriber call is connected, the Session Manager provides the list of rulebase names to the OCS, which chooses the rulebase to be used for the subscriber session from the list.
In case the OCS is not reachable, the rulebase configured as the default will be used.
ECS Deployment and Architecture

The following figure shows a typical example of ECS deployment in a mobile data environment.

Figure 5. Deployment of ECS in a Mobile Data Network

The following figure depicts the ECS architecture managed by the Session Controller (SessCtrl) and Session Manager (SessMgr) subsystems.
Figure 6. ECS Architecture

- CLI
- EMS

ASR 5x00 Management Components

Session Controller

Session Manager (ACS functionality)

Demux Managers

Data Path Components

Event Log Facility

AAA Managers

CDR Subsystem

Event Components

SCT

SIT/RCT

RM

ASR 5x00 Chassis Components
Enhanced Features and Functionality

This section describes enhanced features supported in ECS.

Important: The features described in this section may be licensed Cisco features. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the Managing License Keys section of the Software Management Operations chapter in the System Administration Guide.

Session Control in ECS

In conjunction with the Cisco ASR 5x00 chassis, the ECS provides a high-level network flow and bandwidth control mechanism in conjunction with the Session Control subsystem. ECS Session Control feature uses the interaction between SessMgr subsystem and Static Traffic Policy Infrastructure support of the chassis to provide an effective method to maximize network resource usage and enhancement of overall user experience.

This feature provides the following functionality:

- **Flow Control Functionality**—Provides the ability to define and manage the number of simultaneous IP-based sessions and/or the number of simultaneous instances of a particular application permitted for the subscriber.

  If a subscriber begins a packet data session and system is either pre-configured or receives a subscriber profile from the AAA server indicating the maximum amount of simultaneous flow for a subscriber or an application is allowed to initiate. If subscriber exceeds the limit of allowed number of flows for subscriber or type of application system blocks/redirect/discard/terminate the traffic.

  The following type of flow quotas are available for Flow Control Functionality:
  - **Subscriber-Level Session Quota**—Configurable on a per-rulebase basis
  - **Application-Level Session Quota**—Configurable on a per-charging-action basis

- **Bandwidth Control Functionality**—Allows the operator to apply rate limit to potentially bandwidth intensive and service disruptive applications.

  Using this feature the operator can police and prioritize subscribers’ traffic to ensure that no single or group of subscribers’ traffic negatively impacts another subscribers’ traffic.

  For example, if a subscriber is running a peer-to-peer (P2P) file sharing program and the system is pre-configured to detect and limit the amount of bandwidth to the subscriber for P2P application. The system gets the quota limit for bandwidth from PDP context parameter or individual subscriber. If the subscriber’s P2P traffic usage exceeds the pre-configured limit, the Session Control discards the traffic for this subscriber session.

  Session Control feature in ECS also provides the controls to police any traffic to/from a subscriber/application with the chassis.

Service Group QoS Feature

The Service Group QoS feature enables the chassis/PCEF to define and enforce Fair-Usage-Policy (FUP) per subscriber. This enables changing certain charging-action parameters and all QoS-group-of-ruledefs parameters over the Gx interface per individual subscriber session.
Enhanced Charging Service Overview

In the chassis/PCEF, the Service Group QOS feature enables to:

- Define service-groups that may include unrelated services defined on the chassis/PCEF.
- Dynamically install pre-defined service-groups for a subscriber over Gx.
- Dynamically remove pre-defined service-groups for a subscriber over Gx.
- Dynamically set and change QoS parameters of a service-group for a subscriber over Gx using CCA and RAR messages. QoS parameters of a service-group (FUP-QoS) are:
  - Flow-Rate
  - Flow-Status
  - Volume Threshold
- Apply Flow-Status to a packet-flow progressively at service and service-group levels. The rules for hierarchical enforcement of Flow-Status rule are:
  - Flow-Status Gating at Service Level: If rule indicates “block” or “redirect”, then that action is taken. If rule indicates “allow”, then next level's gating rule is applied.
  - Flow-Status Gating at Service-Group Level: If rule indicates “block” or “redirect”, then that action is taken. If rule indicates “allow”, then next level's gating rule is applied.
- Apply Flow-Rate to a packet-flow progressively at service and service-group levels. Maximum Bit-rate and Burst size is defined by Flow-Rate. Meter the traffic to the configured Flow-Rate and based on the output, apply DSCP marking to the packet.

**Important:** The output action of Flow-Rate can be forward, drop, or mark DSCP. Flow-Rate may allow the packet without DSCP marking.

The rules of hierarchical QoS enforcement are:

- Metering at Service Level: Initially, traffic is metered against service-level QOS rule. If the result of metering marks or drops the packet, then the next level metering is not performed.
- Metering at Service-Group Level: If the packet is allowed at service level, then service-group level QOS metering is done. If the result of metering marks or drops the packet, then the next level metering is not performed.

**Important:** The packet is first subjected to Flow-Status enforcement and if allowed by Flow-Status only then Flow-Rate is enforced. Flow-Status enforcement includes applying Flow-Status progressively at service and service-group Levels. If the flow-status at both levels allows the packet to pass only then it is given for flow-rate enforcement, which applies Flow-Rate progressively at service and service-group levels.

- Monitor volume usage of a group-of-services. Multiple group-of-services can share a volume-quota.
- Provide a mechanism to share configured volume threshold of a service-group across all services in that group. This sharing would be dynamic, that is no predefined quota is allocated per service in a service group.
- Generate a notification to PCRF in a CCR-U message, when volume threshold for a group-of-services is crossed. Once a notification is generated, the trigger is disarmed to generate notification. Continue to monitor usage, but do not report further breaches until PCRF explicitly enables threshold-breath notification trigger in a CCA-U message.
- Report volume usage to PCRF in a CCR-U message when the service-group removed is the last using the shared volume-quota.
Configuration Overview

QoS-group-of-ruledefs are statically configured in the CLI, in the Active Charging Service Configuration Mode. The CLI allows addition and removal of charging and dynamic ruledefs to a named QoS-group-of-ruledefs. A single ruledef can belong to multiple QoS-groups. A maximum of 64 QoS-group-of-ruledefs can be configured in the ACS service. Each QoS-group-of-ruledefs can contain up to 128 ruledefs.

PCRF will be aware of all QoS-group-of-ruledefs names and their constituent ruledefs configured on the chassis/PCEF. The PCRF can activate and remove QoS-group-of-ruledefs for a subscriber session over Gx using a proprietary AVP in CCA and RAR messages. This AVP specifies the name of the QoS-group-of-ruledefs to activate or to remove. Individual ruledefs cannot be dynamically added or removed from a predefined QoS-group-of-ruledefs over the Gx interface. Attributes of QoS-group-of-ruledefs (FUP-QoS parameters) cannot be defined in the CLI. These parameters can only be set and changed over the Gx interface. This feature allows setting different QoS parameters for different subscribers for a named QoS-group-of-ruledefs.

The following attributes of QoS-group-of-ruledefs are supported:

- Precedence or Priority: Priority of a QoS-group-of-ruledefs implies priority of applying QoS-parameters of a QoS-group-of-ruledefs to an incoming data packet. If a packet matches a charging rule which is part of multiple QoS-groups activated for the session, then QoS-parameters of the QoS-group-of-ruledefs with highest priority is applied to the packet. A lower priority number indicates higher priority of application of QoS-parameters of that group. Priority of a QoS-group-of-ruledefs is set by PCRF over Gx for each subscriber session.

- Flow-Status: Can be set to Forward, Block, or Redirect.

  **Important:** The Append-Redirect option is not supported.

  **Important:** Block can be for uplink, downlink, or both uplink and downlink traffic.

- Flow-Rate: Specifies max rate, max burst-size, conform action, and exceed action; individually for uplink and downlink traffic.

- Usage Monitoring Key: A monitoring key, which has an integer value, is set by PCRF over Gx. Volume threshold values are set for this key by PCRF, to perform usage monitoring. Usage is tracked against a monitoring key.

- Volume Thresholds: The PCRF can set volume threshold values for a monitoring key over Gx. An event is reported when thresholds are crossed, and usage is reported at predefined events — such as session termination and when the QoS-group-of-ruledefs removed is the last using the shared volumequota.

  **Important:** In this release, time thresholds are not supported.

- Attributes of QoS-group-of-ruledefs cannot be defined using CLI. These attributes can only be set and changed over Gx. This allows setting different QoS parameters for different subscribers for a named QoS-group-of-ruledefs.

- When a QoS-group-of-ruledefs is activated, its QoS parameters can be set and changed over Gx. This is achieved using a combination of standard and proprietary AVPs.

- The following attributes of charging-action can be set and changed by PCRF over Gx.
  - Flow-Status: Can be set to Forward, Block, or Redirect.
**Important:** ECS does not support the Append-Redirect option.

**Important:** Block can be for uplink, downlink, or both uplink and downlink traffic.

- Flow-Rate: Specifies max rate, max burst-size, conform action, and exceed action; individually for uplink and downlink traffic.
- Volume Threshold: Thresholds are set for usage monitoring. PCRF can set threshold for a monitoring key, which is statically defined for a charging-action using CLI. Usage is reported when thresholds are crossed, and at predefined events such as session termination and removal of QoS-group-of-ruledefs.

**Important:** Monitoring-key is not received over Gx for static charging-action. Triggers for threshold breached are same as the usage-reporting for static charging-action.

- Flow-Status and Flow-Rate can be statically defined for a charging action, and thus applied to a ruledef. These parameters may be over-ridden by PCRF over Gx. Volume-Threshold-Key (monitoring key) can be statically defined for a ruledef in a rulebase. However, its value — the volume quota — can only be set over Gx. Parameters set over Gx will always take precedence over any static configuration.

**Important:** Time-Monitoring over Gx is not supported in this release.

### Support for Service-based QoS

As explained earlier, a service can be mapped in ECS to a set-of-ruledefs with the same charging-action applied to them. This section explains the support for QoS control at the charging-action level:

- Flow-Status: In ECS, you can configure a flow-action in a charging-action. If flow-action is not configured for a charging-action, it implies “Forward” action.

**Important:** Flow-Status value of “Append-Redirect” is not supported by ECS.

- Flow-Rate: ECS charging-action supports configuration of bandwidth limits for a flow. Flow limits can be separately configured for uplink and downlink. ECS supports configuration of peak data-rate and burst-size as well as committed data-rate and burst-size, along with corresponding exceed actions. Specification of committed rate and burst-size is optional.

ECS does not support specifying conform-action (i.e. conform-action is always “Allow”). For exceeding traffic it supports only “Drop” and “Set IP-TOS to 0” as actions. In ECS, traffic matching a flow — both conforming and exceeding, cannot be marked with a specific DSCP mark.

In ECS, charging-action also contains a Content-Id. Multiple charging-actions can contain the same Content-Id. ECS supports a bandwidth-limiting meter per charging action per subscriber session. This metering is separate from traffic meters that are keyed on Content-Id.

- Volume Thresholds: ECS supports setting and monitoring Volume Threshold per flow using the “monitoring-key” mechanism. Monitoring-key is specified in a rulebase configuration. Monitoring-key is associated with a volume-threshold, which is set over Gx. A single monitoring-key can be specified for multiple ruledefs. This allows sharing of assigned volume quota across all the ruledefs with the same Monitoring-Key ID. To
configure service-level volume quota, you can configure the same monitoring-key for all ruledefs that share the same charging action. Monitoring-Key mechanism enables setting and changing Volume-threshold over Gx.

In ECS, changing QoS parameters at a service level means changing parameters of a charging-action.

ECS supports three different kinds of ruledefs:

- Static rules that are defined in the CLI, and are active immediately after they are defined.
- Pre-defined rules that are defined in the CLI and activated/deactivated over Gx.
- Dynamic rules which are defined, activated and deactivated over Gx.

For static and predefined rules, ECS supports updating per-subscriber FUP parameters of a charging-action over Gx. This is achieved using the Charging-Action-Install AVP. Changes to FUP-parameters of dynamic rules are done using the 3GPP-standard Charging-Rule-Definition AVP.

**Hierarchical Enforcement of QoS Parameters**

When a packet arrives, ECS performs Deep Packet Inspection and rule matching. If the packet matches a rule, Control-Charge processing is performed as defined by the matched rule. Ruledef-level and QoS-group-of-ruledefs level QoS enforcement are performed as part of Control-Charge processing.

It is not mandatory to set QoS parameters for a ruledef over Gx. If QoS parameters are not set over Gx, then static definition, if any, is enforced. Similarly, for a subscriber session it is not mandatory to group ruledefs in one or more QoS-group-of-ruledefs. A subscriber may not have any QoS-group-of-ruledefs configured. Incoming traffic may match a ruledef, which has no associated QoS-group-of-ruledef for that subscriber session. In that case, action is taken based only on the configuration for that ruledef.

**Applying Flow-Status**

Flow-Status is applied in a hierarchical manner with the following precedence:

1. Flow Gating at charging-action Level: If flow-action in charging rule indicates “block” or “redirect”, then that action is taken. If rule indicates “allow”, then next level’s gating rule is applied.

**Applying Flow-Rate**

Hierarchy of metering and marking packet follows the precedence:

1. Metering at Charging-Action Level: Flow-Rate at ruledef level is specified in the charging-action associated with the ruledef. Bandwidth metering specified for the charging-action is first applied to every packet. If the packet conforms to specified bandwidth limits, then QoS-group-of-ruledefs level metering will be performed. If the packet exceeds bandwidth limit at charging-action, then specified exceed action will be taken and bandwidth metering at QoS-group-of-ruledefs and subscriber level will not be performed.
2. Metering at QoS-Group-of-Ruledefs Level: If a packet conforms to charging-action bandwidth limits, then QoS-group-of-ruledefs level bandwidth metering will be done. If the packet conforms to specified bandwidth limits, then subscriber-level metering will be performed. If the packet exceeds bandwidth limit at QoS-group, then specified exceed action will be taken.

**Monitoring Usage and Reporting Threshold Breaches**

Volume usage is tracked at the charging-action level and at QoS-group-of-ruledefs level. If a received packet causes volume threshold to exceed, then a trigger ECS sends a CCR-U message to PCRF with Service-Group-Event AVP.
indicating the relevant threshold that was crossed. ECS will then disarm the trigger. If the trigger needs to be rearmed, PCRF will explicitly enable it in the CCA-U message.

**Important:** In 14.0 and later releases, Time Reporting over Gx is supported.

### FUP Enforcement for Dynamic Rules

The chassis/PCEF supports dynamic rule installation using 3GPP-standards-based AVPs. The Charging-Rule-Definition AVP is used to install dynamic rules and configure charging behavior and QoS parameters. For dynamic rules, charging-action is part of the rule definition, and not a separate named entity. QoS parameters of a dynamic-rule are changed using the same Charging-Rule-Definition AVP. For dynamic rules, per-service QoS control maps to per-dynamic-rule QoS-control.

- For dynamic rules, service-level QoS control is supported using 3GPP-Standard AVPs. For hierarchical enforcement of FUP parameters for a packet matching a dynamic rule, charging-action level parameters are read from the dynamic rule itself. Hierarchical FUP enforcement will otherwise be similar to that for predefined rules.
- Dynamic rule has a name associated with it. This name can be added to statically (CLI) defined QoS-group-of-ruledefs. So, a dynamic rule can be configured to be part of a QoS-group-of-ruledefs. Multiple dynamic rules can be part of a QoS-group-of-ruledefs. QoS control for a QoS-group-of-ruledefs is transparently enforced, irrespective of whether constituent ruledefs are static, predefined, or dynamically installed.

### Reporting Statistics and Usage to PCRF

The PCEF reports volume usage to the PCRF in CCR-U and RAR messages at the following events:

- Volume threshold for a charging-action is crossed, and an event trigger for that threshold breach is set by the PCRF.
- Volume threshold for a QoS-group-of-ruledefs is crossed, and an event trigger for that threshold breach is set by the PCRF.
- A QoS-group-of-ruledefs removed is the last using the shared volume-quota.

Monitoring and reporting of time-usage is not supported in this release. Also, packet drops due to enforcement of FUP-QoS parameters is not reported in CDR.

Statistics pertaining to FUP enforcement are available through Show CLI commands for all active sessions.

### Time and Flow-based Bearer Charging in ECS

ECS supports Time-based Charging (TBC) to charge customers on either actual consumed time or total session time usage during a subscriber session. TBC generates charging records based on the actual time difference between receiving the two packets, or by adding idle time when no packet flow occurs.

ECS also supports Flow-based Charging (FBC) based on flow category and type.

PDP context charging allows the system to collect charging information related to data volumes sent to and received by the MS. This collected information is categorized by the QoS applied to the PDP context. FBC integrates a Tariff Plane Function (TPF) to the charging capabilities that categorize the PDP context data volume for specific service data flows.

Service data flows are defined by charging rules. The charging rules use protocol characteristics such as:

- IP address
• TCP port
• Direction of flow
• Number of flows across system
• Number of flows of a particular type

FBC provides multiple service data flow counts, one each per defined service data flow. When FBC is configured in the ECS, PDP context online charging is achieved by FBC online charging using only the wildcard service data flow.

When further service data flows are specified, traffic is categorized, and counted, according to the service data flow specification. You can apply wildcard to service data flow that do not match any of the specific service data flows.

The following are the chargeable events for FBC:

• **Start of PDP context**—Upon encountering this event, a Credit Control Request (CCR) starts, indicating the start of the PDP context, is sent towards the Online Charging Service. The data volume is captured per service data flow for the PDP context.

• **Start of service data flow**—An interim CCR is generated for the PDP context, indicating the start of a new service data flow, and a new volume count for this service data flow is started.

• **Termination of service data flow**—The service data flow volume counter is closed, and an interim CCR is generated towards the Online Charging Service, indicating the end of the service data flow and the final volume count for this service data flow.

• **End of PDP context**—Upon encountering this event, a CCR stop, indicating the end of the PDP context, is sent towards the Online Charging Service together with the final volume counts for the PDP context and all service data flows.

• **Expiration of an operator configured time limit per PDP context**—This event triggers the emission of an interim CCR, indicating the elapsed time and the accrued data volume for the PDP context since the last report.

• **Expiration of an operator configured time limit per service data flow**—The service data flow volume counter is closed and an interim CCR is sent to the Online Charging Service, indicating the elapsed time and the accrued data volume since the last report for that service data flow. A new service data flow container is opened if the service data flow is still active.

• **Expiration of an operator configured data volume limit per PDP context**—This event triggers the emission of an interim CCR, indicating the elapsed time and the accrued data volume for the PDP context since the last report.

• **Expiration of an operator configured data volume limit per service data flow**—The service data flow volume counter is closed and an interim CCR is sent to the Online Charging Service, indicating the elapsed time and the accrued data volume since the last report for that service data flow. A new service data flow container is opened if the service data flow is still active.

• **Change of charging condition**—When QoS change, tariff time change are encountered, all current volume counts are captured and sent towards the Online Charging Service with an interim CCR. New volume counts for all active service data flows are started.

• **Administrative intervention** by user/service also force trigger a chargeable event.

The file naming convention for created xDRs (EDR/UDR/FDRs) are described in the Impact on xDR File Naming section.

### Content Filtering Support

ECS provides off-line content filtering support and in-line static and dynamic content filtering support to control static and dynamic data flow and content requests.
Content Filtering Server Group Support

ECS supports external Content Filtering servers through Internet Content Adaptation Protocol (ICAP) implementation between ICAP client and Active Content Filter (ACF) server (ICAP server).

ICAP is a protocol designed to support dynamic content filtering and/or content insertion and/or modification of Web pages. Designed for flexibility, ICAP allows bearer plane nodes such as firewalls, routers, or systems running ECS to interface with external content servers such as parental control (content filtering) servers to provide content filtering service support.

In-line Content Filtering Support

Content Filtering is a fully integrated, subscriber-aware in-line service available for 3GPP and 3GPP2 networks to filter HTTP and WAP requests from mobile subscribers based on the URLs in the requests. This enables operators to filter and control the content that an individual subscriber can access, so that subscribers are inadvertently not exposed to universally unacceptable content and/or content inappropriate as per the subscribers’ preferences. Content Filtering uses Deep Packet Inspection (DPI) capabilities of ECS to discern HTTP and WAP requests.

**Important:** For more information on Content Filtering support, refer to the Content Filtering Services Administration Guide.

DNS Snooping

This section provides an overview of the DNS Snooping feature.

**Important:** In the 12.2 release, the DNS Snooping feature is supported only on the GGSN and P-GW.

ECS, using L7 rules, can be configured to filter subscriber traffic based on domain name. While this works fine for HTTP-based traffic, a subscriber's initial HTTP request may result in additional flows being established that use protocols other than HTTP and/or may be encrypted. Also, a domain may be served by multiple servers, each with its own IP address. This means that using an IP rule instead of an HTTP rule will result in multiple IP rules, one for each server “behind” the domain. This necessitates service providers to maintain a list of IP addresses for domain-based filters.

The DNS Snooping feature enables a set of IP rules to be installed based on the response from a DNS query. The rule in this case contains a fully qualified domain name (for example, m.google.com) or its segment (for example, google) and a switch that causes the domain to be resolved to a set of IP addresses. The rules installed are thus IP rules. Any actions specified in the domain rule are inherited by the resulting IP rules.

When configured, DNS snooping is done on live traffic for every subscriber.

The DNS Snooping feature enables operators to create ruledefs specifying domain names or their segments. On defining the ruledefs, the gateway will monitor all the DNS responses sent towards the UE, and snoop only the DNS response that has q-name or a-name as specified in the rules, and identify all the IP addresses resulting from the DNS response. A table of these IP addresses is maintained per destination context per rulebase per instance and shared across subscribers of the same destination context same rulebase per instance. In case DNS queries made by different subscribers produce different results, all the IP entries in the table are stored based on their Time to Live (TTL) and the configurable timer. The TTL or the timer whichever is greater is used for aging out the IP entry. Dynamic IP rules are created for these IP entries within the same rule having the domain name, applying the same charging action to these dynamic rules. This solution will have the exact IP entries as obtained live from snooping DNS responses. They will be geographically and TTL correct.
License Requirements

DNS Snooping is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the Managing License Keys section of the Software Management Operations chapter in the System Administration Guide.

Bulkstatistics Support

Bulkstatistics reporting for the DNS Snooping feature is supported.
For the DNS Snooping feature the following bulkstatistics are available in the ECS schema:

- ecs-dns-learnt-ipv4-entries
- ecs-dns-flushed-ipv4-entries
- ecs-dns-replaced-ipv4-entries
- ecs-dns-overflown-ipv4-entries
- ecs-dns-learnt-ipv6-entries
- ecs-dns-flushed-ipv6-entries
- ecs-dns-replaced-ipv6-entries
- ecs-dns-overflown-ipv6-entries

How it Works

This section describes how the DNS Snooping feature works.

ECS allows operators to create ruledefs specifying domain names or their segments using options available in the CLI ruledef syntax (contains, starts-with, ends with, or equal to). This allows operators to match all the traffic going to specified fully qualified domain names as presented by the UE in the DNS queries, or segments of the domain names.

Internally, when a ruledef containing ip server-domain-name keyword is defined and the ruledef is used in a rulebase, an IP table similar to the following is created per rulebase per instance.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Domain Name</th>
<th>IP Pool Pointer</th>
<th>Associated Ruledef</th>
<th>List of CNAMEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>gmail</td>
<td>ip-pool1</td>
<td>domain_google</td>
<td>l.google.com</td>
</tr>
<tr>
<td>=</td>
<td>yahoo.com</td>
<td>ip-pool2</td>
<td>domain_yahoo</td>
<td></td>
</tr>
<tr>
<td>starts-with</td>
<td>gmail</td>
<td>ip-pool3</td>
<td>domain_start_gmail</td>
<td></td>
</tr>
</tbody>
</table>

On definition of the ruledefs, the gateway will monitor all the DNS responses sent towards the UE and will snoop the DNS responses from valid DNS servers. IP addresses (IPv4 and IPv6) resulting from the DNS responses are learnt dynamically and will be used for further rule matching. These dynamic Service Data Flows (SDFs), containing IP addresses, may also be reused by ECS for other subscribers from the same routing instance in order to classify the subscriber traffic.

The dynamic SDFs generated are kept for the TTL specified in the DNS response plus a configurable timer that can be added to the TTL in case the DNS response contains a very small TTL.
Important: If the rule created using this feature is removed from the configuration then all the associated
dynamic SDFs are removed immediately. The usage incurred by the subscriber for traffic matching the removed SDFs
will be reported over the Gy interface when the usage reporting for the corresponding rating group is due.

In case DNS queries made by different subscribers produce different results, all the dynamically generated SDFs are
stored based on their TTL and the configured timer.

DNS Snooping supports DNS responses containing nested CNAME responses.

When the DNS response contains nested CNAME record, a list per entry in the IP-table is dynamically allocated to store
the CNAME. CNAME is the canonical name of the alias, which means the q-name to which the actual query was made
is the alias name and this CNAME is the actual domain name to which the query should be made. So, the IP addresses
found in response to CNAME DNS query is stored in the same IP-pool as that of the alias.

Here, either the DNS response to the actual alias contains CNAME record along with its A record or only the CNAME
record. In the first case the IP address is already resolved for CNAME and it is included in the learnt IP addresses IP-
pool.

In both the scenarios, the list of CNAMEs is stored in the same record of the IP-table, which is keyed by
operator+domain. By default, the operator for CNAME is "equal". So, while snooping DNS responses, DNS responses
for a-name as in the CNAME list will also be snooped and the IP addresses stored in the corresponding IP-pool. This
allows the feature to work in case DNS responses have nested CNAME response.

Like IP addresses, even CNAME entries have TTL associated with them. In the same five minute timer, where the aged
IP addresses are timed out, the CNAME entries will also be looked at and the expired CNAME entries reference
removed from the corresponding entry.

The DNS Snooping feature supports both IPv4 and IPv6 addresses. The following are the maximum limits:

- IPv4 addresses learnt per server-domain-name pattern: 200
- IPv4 addresses learnt per instance across all IPv4 pools: 51200
- IPv6 addresses learnt per server-domain-name pattern: 100
- IPv6 addresses learnt per instance across all IPv6 pools: 25600

Rule matching: While matching rule for IP packets, it will be checked if the source IP address matches any of the entries
stored in the IP pools formed as part of DNS snooping. If a match is found, the corresponding ruledef is determined
from the IP table. The other rule lines of the rule are matched, and if it is the highest priority rule matched it is returned
as a match. The corresponding charging-action is applied. So the same priority as that of the domain name is applied to
its corresponding IP addresses, and is matched as a logical OR of the domain or the IP addresses.

Lookup (matching) is performed in learnt IP pools only for the first packet of the ADS as the destination IP address will
not change for that flow, and will match the same rule (last rule matched for this ADS flow) for all the packets of the
flow. This enables to have the same rule matched even if its IP addresses get aged out when the flow is ongoing.

In 12.3 and earlier releases, the CLI command `show active-charging dns-learnt-ip-addresses
statistics sessmgr all` displayed all the configured patterns and rulebase names for each pattern entry, even
though the pattern has not learnt any IP address.

When a large number of DNS snooping ruledefs are configured (configured as ip server-domain name under ruledef
configuration), the memory allocated for sending this information exceeds the message size limit for messenger calls
and hence the crash is observed.

In 14.0 and later releases, the CLI command `show active-charging dns-learnt-ip-addresses
statistics sessmgr all` will be displaying only the patterns for which at least one IPv4/IPv6 address is learnt as all other
information is available from the configuration.

The following call flow illustration and descriptions explain how the DNS Snooping feature works.
Table 2. DNS Snooping Call Flow Descriptions

<table>
<thead>
<tr>
<th>Step No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UE requests the system for registration.</td>
</tr>
<tr>
<td>2</td>
<td>System processes UE-related information with ECS subsystem.</td>
</tr>
<tr>
<td>3</td>
<td>System sends AAA Access Request to AAA server for UE.</td>
</tr>
<tr>
<td>4</td>
<td>The AAA server processes the AAA Access Request from the ECS to create the session, and the Policy Manager in AAA server uses subscriber identification parameters including NAI (username@domain), Calling Station ID (IMSI, MSID), and Framed IP Address (HoA) as the basis for subscriber lookup.</td>
</tr>
<tr>
<td>Step No.</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 5        | The Policy Manager and AAA generate and send an Access Accept message including all policy and other attributes to establish the session to ECS. The Policy Manager and/or AAA include following attributes in the Access Accept message:  
- Filter ID or Access Control List Name: Applied to subscriber session. It typically contains the name of the Content Service Steering (CSS) ACL. The CSS ACL establishes the particular service treatments such as Content Filtering, ECS, Stateful Firewall, VPN, etc. to apply to a subscriber session, and the service order sequence to use in the inbound or outbound directions. Real-time or delay sensitive flows are directly transmitted to the Internet with no further processing required. In this case, no CSS ACL or Filter ID is included in the Access Response.  
- SN1-Rulebase Name: This custom attribute contains information such as consumer, business name, child/adult/teen, etc. The rulebase name identifies the particular rule definitions to apply. Rulebase definitions are used in ECS as the basis for deriving charging actions such as prepaid/postpaid volume/duration/destination billing and charging data files (EDRs/UDRs). Rulebase configuration is defined in the ACS Configuration Mode and can be applied to individual subscribers, domains, or on per-context basis. |
| 6        | ECS creates a new session for UE, and sends the rulebase to ACS subsystem if required. |
| 7        | ECS sends Accounting-Start messages to the AAA server. |
| 8        | The AAA server sends Accounting-Start response message to ECS. |
| 9        | ECS establishes data flow with UE. |
| 10       | UE requests for data with URL name (DNS query). |
| 11       | ECS analyzes the query-name from the subscriber’s DNS query, and if it matches the entry in the “DNS URLs to be snooped” list (created when ip server-domain-name rules were defined in rulebase), it marks this request for its response to be snooped. |
| 12       | DNS query is sent to the Internet. |
| 13       | DNS response is received from the Internet. |
| 14       | Based on the various answer records in the response the IP addresses are snooped and included in the “list of learnt IP addresses”. |
| 15       | DNS response is sent to the UE. |
| 16       | Actual URL request comes from the UE. |
| 17       | Looking at the server-ip-address of the packet, rule matching will be done based on the “list of learnt IP addresses” and the rules already configured. An action is taken based on the ruledef matched and the charging action configured. |
| 18       | If the packet is to be forwarded, it is forwarded to the Internet. |
| 19       | A response is received from the Internet. |
| 20       | The response is sent to the UE. |
| 21       | UE requests for session termination. |
| 22       | System sends Accounting-Stop Request to AAA server. |
| 23       | AAA server stops accounting for subscriber and sends Accounting-Stop-Response to the system. |
Limitations and Dependencies

This section identifies limitations and dependencies for the DNS Snooping feature.

- On a SessMgr kill or card switchover, the dynamic IP rules created based on domain name resolution will be lost. Until a new DNS query is made, the dynamic IP based rules will not be applied. These rules will be recreated on new DNS traffic. So, SessMgr recovery is not supported for these dynamic IP rules.

- The `ip server-domain-name` ruledef can be used as a predefined dynamic rule, static rule, or as a part of group of ruledefs. However, it cannot be used as a dynamic-only rule, as dynamic-only rules apply up to L4 and this is an L7 rule.

- Operators must define valid domain-name servers, the DNS responses from which will be considered correct and snooped and included in the list of dynamic-learnt IP addresses. If the list of valid domain-name servers is not provided, then the DNS responses from all DNS servers will be considered valid and included in the list of learnt IP addresses. Also, in case subscribers make DNS queries to their self-created DNS servers and hack the response being sent, it can result in inclusion of invalid IP addresses in the list. In this case, the IP addresses will be learnt and the traffic may be free-rated or blocked incorrectly depending on the action set. Therefore the above is suggested to avoid attacks on DNS traffic.

- There is a limit on the total number of learnt IP addresses per server-domain-name ruledef for memory and performance considerations. Any more IP addresses across this limit will not be learnt and hence the charging-action will not be applied to these IP addresses. Similarly, there is a limit on the total number of server-domain-name ruledefs that can be configured.

- If same IP address is returned in DNS responses for different DNS q-names (same IP hosting multiple URLs), than while rule matching, the higher priority rule having this learnt-IP address will be matched. This can have undesired rule-matching as explained next.

For example, if DNS queries for both www.facebook.com and www.cnn.com returned the IP address 162.168.10.2. Here we have allow action for domain www.facebook.com and block or no action for www.cnn.com which is at a lower priority than allow rule. In this if the actual request for www.cnn.com comes than as the server IP is same, it will match the higher priority allow rule for domain www.facebook.com (considering there are no other rule lines or all lines match) and thus, free rated incorrectly. However, this will happen only of same IP address is returned for different q-names, which is rare and cannot be handled.

- In the 12.2 release, the lookup for IPv6 learnt IP addresses will not be optimized. Hash based lookup (optimization) is done for IPv4 address lookup. In a later release Longest Prefixed Match (LPM) based optimization will be considered for both IPv4 and IPv6 learnt IP address matching.

IP Readdressing

The IP Readdressing feature enables redirecting unknown gateway traffic based on the destination IP address of the packets to known/trusted gateways.

IP Readdressing is configured in the flow action defined in a charging action. IP readdressing works for traffic that matches particular ruledef, and hence the charging action. IP readdressing is applicable to both uplink and downlink traffic. In the Enhanced Charging Subsystem, uplink packets are modified after packet inspection, rule matching, and so on, where the destination IP/port is determined, and replaced with the readdress IP/port just before they are sent out. Downlink packets (containing the readdressed IP/port) are modified as soon as they are received, before the packet inspection, where the source IP/port is replaced with the original server IP/port number.

For one flow from an MS, if one packet is re-addressed, then all the packets in that flow will be re-addressed to the same server. Features like DPI and rule-matching remain unaffected. Each IP address + port combination will be defined as a ruledef.
In case of IP fragmentation, packets with successful IP re-assembly will be re-addressed. However, IP fragmentation failure packets will not be re-addressed.

New hierarchy approach has also been provided for selecting the server in case of server list configured under charging-action. This helps the operator to specify list of DNS servers in the order of preference. In hierarchy based approach, queries are redirected as per primary, secondary, and tertiary selection. Both round-robin and hierarchy based server selection approaches would be applicable for both IPv4 and IPv6 based servers. An additional CLI is provided that enables you to select from hierarchy or round-robin approach for server selection. See the Configuring IP Readdressing for more information.

URL-based Re-addressing

URL-based re-addressing feature is applied based on L7 rule matching for HTTP URLs in addition to re-addressing charging action based on L3/L4 rule matching. HTTP request with specific token or complete URL must be redirected to a separate server and must be transparent to the UE.

Flow-based re-addressed connection

Flow-based re-addressed connection is the default behavior of this feature. In this type, after a HTTP connection is setup with Original Server, all subsequent requests will be sent to it until the URL-based re-addressing rule matches. This behavior holds true even for multiple concatenated HTTP requests in one packet.

The following call flow explains the URL HTTP Request Re-addressing feature.

![Figure 8. URL HTTP Request Re-addressing](image-url)
### Enhanced Features and Functionality

**Table 3. URL HTTP Request Re-addressing**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1—3</td>
<td>UE sets up a TCP connection with the Origin Server (OS) by sending SYN. The TCP three-way handshake takes place between UE and the Origin Server.</td>
</tr>
<tr>
<td>4</td>
<td>UE sends a HTTP request to the OS which passes through the ASR5x00 L7 DPI rule-matching. The URL of the request contains a known token, domain name, or a token configured at ASR5x00 (in a ruledef).</td>
</tr>
<tr>
<td>5—8</td>
<td>ASR5x00 using L7 DPI recognizes that the request is for New Server. ASR5x00 breaks/closes the existing TCP connection with OS and establishes a new connection with the New Server. TCP proxy is used to maintain TCP connection between UE and ASR5x00.</td>
</tr>
<tr>
<td>9</td>
<td>ASR5x00 sends the HTTP Request destined for Origin Server to New Server.</td>
</tr>
<tr>
<td>10</td>
<td>New Server sends back the content in HTTP Response.</td>
</tr>
<tr>
<td>11</td>
<td>ASR5x00 proxies the content back to the UE.</td>
</tr>
<tr>
<td>12</td>
<td>UE closes the TCP connection.</td>
</tr>
<tr>
<td>13</td>
<td>ASR5x00 closes the connection with the New Server.</td>
</tr>
</tbody>
</table>

### Next-hop Address Configuration

ECS supports the ability to set the next-hop default gateway IP address as a charging action associated with any ruledef in a rulebase. This functionality provides more flexibility for service based routing allowing the next-hop default gateway to be set after initial ACL processing. This removes need for AAA to send the next-hop default gateway IP address for CC opted in subscribers.

In 15.0 and later releases, ECS behaves such that rule matching is not done for partial HTTP request if HTTP analysis is enabled.

Assume ECS has received partial HTTP GET packet where URL is not complete, and there are a few URL based rules configured. At this point of time, ECS will not be in a position to match proper rule as complete URL information is not available. When packet where request is completed, is received by ECS, proper rule matching is possible. Earlier partial packets and bytes of this request will be charged accordingly.

Also, this does not apply to post-processing rules. Post-processing rules are matched for all the packets, irrespective of the packet is partial or not. If the customer wants to configure actions like next-hop forwarding or ip-readdressing, then that can be configured in post-processing rules.

In releases prior to 15.0, partial packets do not go for post processing rule match. Whereas in 15.0 and later releases, the partial packets go for required rule match. This behavior change is introduced to obtain the correct statistics about the packets.

**How it works:**

**Step 1** The next-hop address is configured in the charging action.

**Step 2** Uplink packet sent to ECS is sent for analysis.

**Step 3** When the packet matches a rule and the appropriate charging action is applied, the next-hop address is picked from the charging action and is copied to the packet before sending the packet to Session Manager.
Step 4  Session Manager receives the packet with the next-hop address, and uses it accordingly.

**Post Processing**

The Post Processing feature enables processing of packets even if the rule matching for them has been disabled. This enables all the IP/TCP packets including TCP handshaking to be accounted and charged for in the same bucket as the application flow. For example, delay-charged packets for IP Readdressing and Next-hop features.

- Readdressing of delay-charged initial hand-shaking packets.
- Sending the delay-charged initial packets to the correct next-hop address.
- DCCA—Taking appropriate action on retransmitted packets in case the quota was exhausted for the previous packet and a redirect request was sent.
  - DCCA with buffering enabled—Match CCA rules, charging-action will decide action—terminate flow/redirect
  - DCCA with buffering disabled—Match post-processing rules, and take action
- Content ID based ruledefs—On rule match, if content ID based ruledef and charging action are present, the rule is matched, and the new charging action will decide the action

A ruledef can be configured as a post-processing rule in the ruledef itself using rule-application of the ruledef. A rule can be charging, routing, or a post-processing rule. If the same ruledef is required to be a charging rule in one rulebase and a post-processing rule in another one, then two separate identical ruledefs must be defined.

How the Post-processing Feature Works

The following steps describe how the Post-processing feature works:

**Step 1**  Charging rule-matching is done on packets and the associated charging-action is obtained.

**Step 2**  Using this charging-action the disposition-action is obtained.

**Step 3**  If the disposition action is to either buffer or discard the packets, or if it is set by the ACF, or if there are no post-processing rules, the packets are not post processed. The disposition action is applied directly on the packets. Only if none of the above conditions is true, post processing is initiated.

**Step 4**  Post-processing rules are matched and the associated charging-action and then the disposition-action obtained through control-charge.

**Step 5**  If both match-rule and control-charge for post processing succeed, the disposition-action obtained from post-processing is applied. Otherwise, the disposition-action obtained from charging rule-matching is used.

If no disposition action is obtained by matching post-processing rules, the one obtained by matching charging-rules will be applied.

Irrespective of whether post processing is required or not, even if a single post-processing rule is configured in the rulebase, post processing will be done.

The following points should be considered while configuring post-processing rules for next-hop/readdressing.

- The rules will be L3/L4 based.
- They should be configured in post-processing rules' charging actions.
Enhanced Charging Service Overview

Enhanced Features and Functionality

For x-header insertion, there should either be a post-processing rule whose charging-action gives no disposition-action or the packet should not match any of the post-processing rules so that the disposition action obtained from charging-rule matching is applied.

Tethering Detection

This section provides an overview of the Tethering Detection feature.

Important: In this release, the Tethering Detection feature is supported only on the GGSN, HA, and P-GW.

Tethering refers to the use of a mobile smartphone as a USB dongle/modem to provide Internet connectivity to PC devices (laptops, PDAs, tablets, and so on) running on the smartphone's data plan. Typically, for smartphone users, most operators have in place an unlimited data plan, the usage of which is intended to be from the smartphone as a mobile device. However, some subscribers use the low cost / unlimited usage data plan to provide Internet connectivity to their laptops in places where normal Internet connection via broadband/WiFi may be costly, unavailable, or insecure.

The Tethering Detection feature enables detection of subscriber data traffic originating from PC devices tethered to mobile smartphones, and also provides effective reporting to enable service providers take business decisions on how to manage such usage and to bill subscribers accordingly.

Important: In the 12.2 release, Tethering Detection is supported only for IPv4 (TCP) traffic flows.

The Tethering Detection feature is enabled on a per rulebase basis. The rulebase (billing plan) assigned for APN will contain the tethering detection related configuration. ECS performs tethering detection on a per flow basis for all subscribers (for whom TAC database match succeeded) using an APN in which the feature is enabled. The extent to which the detection mechanism is executed depends on the type of flow. If it is a non-TCP flow, for example UDP or ICMP, then tethering detection is not possible for the same.

License Requirements

Tethering Detection is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the Managing License Keys section of the Software Management Operations chapter in the System Administration Guide.

MUR Support for Tethering Detection

The ASR chassis works in conjunction with the Mobility Unified Reporting (MUR) application to facilitate tethering detection on the chassis.

If MUR is not deployed, then the database file must be manually placed on the ASR chassis, in the /hd-raid/databases/ directory, and loaded into configuration using CLI command.

For more information on MUR, refer the MUR Online Help System and the Mobility Unified Reporting System Installation and Administration Guide.
Tethering Detection Databases

The Tethering Detection database files must be populated and loaded on to the ASR chassis by the administrator. The procedure to load the database is the same for all the different databases.

Before the database(s) can be loaded for the first time, tethering detection must be enabled using the `tethering-database` CLI command in the ACS Configuration Mode.

For all the databases, only a full upgrade of a database file is supported. Incremental upgrade is not supported. If, for any particular database, the upgrade procedure fails, the system will revert back to the previous working version of that database.

In 15.0 and later releases, os-signatures can be collected from TCP SYN even when tethering detection is disabled in the rulebase. The os-signature will be parsed if an EDR/UDR with an os-signature variable is present in a rulebase or charging-action in the rulebase. This can be used to collect os-signatures that can then be used to build an OS database for the tethering detection feature.

Loading and Upgrading Tethering Detection Databases

This section provides an overview of loading and upgrading the databases used in tethering detection.

The database files from MUR must be copied onto the ASR chassis to the following directory path designated for storing the database files:

```
/hd-raid/databases/
```

Any further upgrades to the database files can be done by placing the file named `new-filename` in the designated directory path. ACS auto-detects the presence of files available for upgrade daily. When a new version of a file is found, the upgrade process is triggered. The upgrade can also be forced by running the upgrade command in the CLI. On a successful upgrade this file is renamed to `filename`.

Session Recovery Support

The following Session Recovery features are supported:

- Database recovery after SessCtrl getting killed.
- Database recovery after one or more SessMgrs getting killed.

Note that depending on the size of the database files and the number of SessMgrs operational in the system, it may take sometime (ranging from five seconds to five minutes) for the database to become available in all the SessMgrs post recovery/migration.

Time-of-Day Activation/Deactivation of Rules

Within a rulebase, ruledefs/groups-of-ruledefs are assigned priorities. When packets start arriving, as per the priority order, every ruledef/group-of-ruledefs in the rulebase is eligible for matching regardless of the packet arrival time. By default, the ruledefs/groups-of-ruledefs are active all the time.

The Time-of-Day Activation/Deactivation of Rules feature uses time definitions (timedefs) to activate/deactivate static ruledefs/groups-of-ruledefs such that they are available for rule matching only when they are active.

**Important:** The time considered for timedef matching is the system’s local time.
How the Time-of-Day Activation/Deactivation of Rules Feature Works

The following steps describe how the Time-of-Day Activation/Deactivation of Rules feature enables charging according to the time of the day/time:

**Step 1**
Timedefs are created/deleted in the ACS Configuration Mode.
A maximum of 10 timedefs can be created in an ECS service.

**Step 2**
Timedefs are configured in the ACS Timedef Configuration Mode. Within a timedef, timeslots specifying the day/time for activation/deactivation of rules are configured.
A maximum of 24 timeslots can be configured in a timedef.

**Step 3**
In the ACS Rulebase Configuration Mode, timedefs are associated with ruledefs/groups-of-ruledefs along with the charging action.
One timedef can be used with several ruledefs/group-of-ruledefs. If a ruledef/group-of-ruledefs does not have a timedef associated with it, it will always be considered as active.

**Step 4**
When a packet is received, and a ruledef/group-of-ruledefs is eligible for rule matching, if a timedef is associated with the ruledef/group-of-ruledefs, before rule matching, the packet-arrival time is compared with the timeslots configured in the timedef. If the packet arrived in any of the timeslots configured in the associated timedef, rule matching is undertaken, else the next ruledef/group-of-ruledefs is considered.

This release does not support configuring a timeslot for a specific date.
If, in a timeslot, only the time is specified, that timeslot will be applicable for all days.
If for a timeslot, “start time” > “end time”, that rule will span the midnight. That is, that rule is considered to be active from the current day until the next day.
If for a timeslot, “start day” > “end day”, that rule will span over the current week till the end day in the next week.

In the following cases a rule will be active all the time:
- A timedef is not configured in an action priority
- A timedef is configured in an action priority, but the named timedef is not defined
- A timedef is defined but with no timeslots

Transactional Rule Matching

This section provides an overview of the transactional rule matching feature.

The Transactional Rule Matching (TRM) feature enables the Enhanced Charging Service (ECS) to bypass per-packet rule matching on a transaction once the transaction is fully classified. This enables ECS to better utilize CPU resources and accommodate additional throughput for the system, thus improving the overall performance.

A transaction for TRM can be defined as the entire UDP flow, the ACK of the 3-way handshake to the FIN/RST of a TCP flow, or the HTTP request to the next HTTP request, or HTTP request to the FIN/RST for the final request of the flow. The TRM feature can perform rule matching on IP L4 rules (UDP, TCP), HTTP, and HTTPS.

Fastpath

The Fastpath feature can be used to reduce the overall system performance impact as a large amount of data packet is consumed through the ECS data path. The Fastpath feature introduces an alternate ECS data path (Fastpath) with limited
supported features. By limiting the supported features, Fastpath eliminates the overhead of packets being subjected to the large number of feature-based conditional checks in ECS.

Fastpath does not replace the existing data path, and works in parallel with the existing ECS data path. The Fastpath feature is part of the Transactional Rule Matching (TRM) feature and requires TRM to be enabled.

**Important:** From 16.0 release, **Transactional Rule Matching** and **Fastpath** functionalities have been merged, and will be governed by only the `transactional-rule-matching` keyword alone. The keyword `fastpath` independently can no longer be used to turn on or turn off this functionality.

### Feature Support

The following table provides information on the supported and unsupported features of Fastpath. Features that are listed under the Optimized column in the table below indicate that the features are directly supported by Fastpath. Features that are listed under the Eligible column in the table below indicate that a flow requiring the feature does not prevent it from taking advantage of Fastpath. Features that are listed under the unsupported column in the table below indicate that all packets in the flow that belong to the feature is not supported for Fastpath and will take the existing ECS data path.

**Important:** The TRM feature is supported in SSI platform; earlier it was restricted only to ASR5500.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Fastpath Optimized</th>
<th>Fastpath Eligible</th>
<th>Unsupported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Url Redirect</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Charging Bucket Maintenance</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>ITC/BW control</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Next Hop</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>TCP State based rules</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Post-processing Rules</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Flow limit - Discard/Redirect/Terminate-Flow/Terminate-Session</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>DSCP / IP TOS</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>ICSR</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Session Recovery</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Content Filtering(CF) Static</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>CF Dynamic</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Socket Migration</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Blacklisting</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>ICAP</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>NAT</td>
<td>Yes</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
### Enhanced Features and Functionality

<table>
<thead>
<tr>
<th>Feature</th>
<th>Fastpath Optimized</th>
<th>Fastpath Eligible</th>
<th>Unsupported</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFW</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Video transrating</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>MVG CAE Readdressing</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>MVG Pacing</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>MVG Link Monitoring</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>MVG Header Insertion</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>P2P</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>SIP-ALG (App Layer Gateway)</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>H323 - ALG</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>DCCA</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>IPv6</td>
<td>Yes</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Flow Readdress</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Idle-timeout handling</td>
<td>Yes</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Connection termination (2MSL)</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>TCP Proxy</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>QOS</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Lawful Intercept</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>HTTP/HTTPS</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Non HTTP L7 protocols</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>NON UDP/TCP flows</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Tethering detection</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Gx</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Gy</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>HEE</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Radius</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Diameter</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>L4 checksum</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>TCP link monitoring</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Header enrichment</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Wimax Hotlining</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Parsing Error Detection Denial</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>IP only Byte Counting/Charging</td>
<td>Yes</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Even when a flow is supported for Fastpath, some packets for the flow are not eligible to be processed in Fastpath. When a packet is not eligible, the packet is processed in the existing ECS data path. The following table provides information on the packet-level support in Fastpath:

**Table 5. Packet-level Support**

<table>
<thead>
<tr>
<th>Packet Handling Feature</th>
<th>Fastpath Eligible</th>
<th>No Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid UDP/TCP in order pkts</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>OOO Packet handling</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>TCP Retransmissions</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>IP Fragmentation</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>TCP Handshaking</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>TCP Termination</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>First packet of Flow</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Gx Rule Update</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Invalid L3/L4 packet</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Packet already queued</td>
<td>—</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Bulkstatistics Support**

Bulkstatistics reporting for the TRM feature is supported.

The following bulkstatistics are available in the ECS schema:

- `trm-rule-match-bypassed`
- `trm-rule-match-bypass-triggered`
- `fp-eligible-flows`
- `fp-packets`
- `fp-failures`
Limitations and Dependencies

The following are limitations to the TRM feature:

- TRM is supported only on the ASR5500 platform.
- TRM is limited to flows with no protocol routing rules with the exception of HTTP and HTTPS flows. All other flows are not supported and TRM does not have any impact on other flows.
- When TRM is enabled, the following functionalities are affected:
  - Per direction rule matching.
  - TCP state rules for the duration of the TRM transaction.
  - Configuring delay charging when the TRM feature is enabled impacts only packets outside transaction boundaries. All packets within the transaction boundary will be applied to the application (i.e. HTTP).
- Once a flow is classified to a ruledef (first packet in flow for UDP, or the first data packet after the 3-way handshake for a TCP flow), TRM will attempt to use that matched rule for the duration of the transaction. This might result in the ruledefs such as those with mid-transaction TCP states or packet direction to be ignored for the flow.

Support for WebSocket Protocol Identification

This feature extends support for WebSocket Protocol identification.

The WebSocket protocol is an independent TCP based protocol. A connection is identified as WebSocket through the first HTTP Get Request header after the three way handshake. This packet includes an upgrade header (Upgrade: websocket) and other WebSocket headers (Sec-WebSocket-*) to upgrade HTTP to WebSocket protocol. This helps operators to categorize WebSocket traffic and apply different policies for such traffic.

A new CLI “websocket flow-detection” has been implemented at rulebase level to detect the WebSocket protocol. The WebSocket protocol identification can be enabled or disabled with the new CLI WebSocket protocol.

For more information on these commands, see the ACS Rulebase Configuration Mode Commands chapter in the Command Line Interface Reference guide.

How it Works

This section describes how WebSocket Protocol Identification feature works.

If the WebSocket detection is enabled in the rulebase, the ECS parser looks for the following fields in the HTTP Get header fields Host, Upgrade, Connection, Sec-WebSocket-Key, Origin, and Sec-WebSocket-Version. If these headers are present, the TCP connection is upgraded to a WebSocket connection. A ruledef can be defined to identify the HTTP GET request for the websocket and rate it in a certain way. The subsequent data that is transferred through the websocket is also billed the same way as the first packet.

Important: You need to enable HTTP analysis to detect Websockets, and WebSocket connections cannot be detected on secure-HHTTP connections.

URL Filtering

The URL Filtering feature simplifies using rule definitions for URL detection.
The following configuration is currently used for hundreds of URLs:

```
ruledef HTTP://AB-WAP.YZ

www url starts-with HTTP://CDAB-SUBS.OPERA-MINI.NET/HTTP://AB-WAP.YZ

www url starts-with HTTP://AB-WAP.YZ

multi-line-or all-lines

exit
```

In the above ruledef:

- The HTTP request for the URL “http://ab-wap.yz” is first sent to a proxy “http://cdab-subs.opera-mini.net/”.
- The URL “http://cdab-subs.opera-mini.net/” will be configured as a prefixed URL.

Prefixed URLs are URLs of the proxies. A packet can have a URL of the proxy and the actual URL contiguously. First a packet is searched for the presence of proxy URL. If the proxy URL is found, it is truncated from the parsed information and only the actual URL (that immediately follows it) is used for rule matching and EDR generation.

The group-of-ruledefs can have rules for URLs that need to be actually searched (URLs that immediately follow the proxy URLs). That is, the group-of-prefixed-URLs will have URLs that need to be truncated from the packet information for further ECS processing, whereas, the group-of-ruledefs will have rules that need to be actually searched for in the packet.

URLs that you expect to be prefixed to the actual URL can be grouped together in a group-of-prefixed-URLs. A maximum of 64 such groups can be configured. In each such group, URLs that need to be truncated from the URL contained in the packet are specified. Each group can have a maximum of 10 such prefixed URLs. By default, all group-of-prefixed-URLs are disabled.

In the ECS rulebase, you can enable/disable the group-of-prefixed-URLs to filter for prefixed URLs.


### Implementation of AES Encryption

URL redirection is used for user equipment (UE) self-activation, along with pre-paid mobile broadband and other projects.

In the current implementation, when a URL redirection occurs, additional dynamic fields such as MSISDN, IMEI, and username can be appended to the redirection URL for use by the IT portal during the account activation process. StarOS currently supports URL encryption of attributes within the redirection by using Blowfish (64 and 128 bit keys) encryption. It also provides the ability to encrypt either single or multiple concatenated plain text fields. However, Blowfish is no longer considered robust and thus operator now has the option to augment the security of these redirection parameters with a more robust encryption based on AES Encryption.

For URL encryption, AES is an additional option along with Blowfish. The operator has flexibility of choosing the encryption mechanism— Blowfish or AES. This is achieved using CLI and there are no changes done to the dynamic fields. The operator can have different encryption for different rules configurable using CLI.
AES encryption is available for 128 and 256 bit keys. For AES encryption with CBC mode of operation, a key-phrase is taken as configurable field from the operator. This key phrase is internally converted to a 128/256 bit key. An additional field value ("salt") is also allowed as a configurable field. This configurable field is optional.

Security of the subscriber sensitive attributes is enhanced with a more robust encryption algorithm. This helps protect subscriber specific information sent to different servers, thus helping operators to adhere to regulatory policies.

For more information on these commands, see the ACS Charging Action Configuration Mode Commands chapter in the Command Line Interface Reference.

TCP Proxy

The TCP Proxy feature enables the ASR 5x00 to function as a TCP proxy. TCP Proxy is intended to improve ECS subsystem’s functionality in case of Content Filtering, ICAP, RADIUS Prepaid, Redirection, Header Enrichment, Stateful Firewall, Application Detection and Control, DCCA, and Partial Application Headers features.

TCP Proxy along with other capabilities enables the ASR 5x00 to transparently split every TCP connection passing through it between sender and receiver hosts into two separate TCP connections, and relay data packets from the sender host to the receiver host via the split connections. This results in smaller bandwidth delay and improves TCP performance.

The TCP Proxy solution comprises of two main components:

- User-level TCP/IP Stacks — The TCP Proxy implementation uses two instances of the User Level TCP/IP stack. The stack is integrated with ECS and acts as packet receiving and sending entity. These stacks modify the behavior in which the connection is handled.

- Proxy Application — The Proxy application binds ECS, stack, and all the applications. It is the only communicating entity between the two stacks and the various applications requiring the stack. The TCP Proxy application manages the complete connection. It detects connection request, connection establishment, and connection tear-down, and propagates the same to the applications. Whenever the buffers are full, the Proxy application also buffers data to be sent later.

On an ASR 5x00 chassis, the TCP Proxy functionality can be enabled or disabled and configured from the CLI, enabling the ASR 5x00 to perform either in proxy or non-proxy mode. TCP Proxy can either be enabled for all connections regardless of the IP address, port, or application protocol involved, or for specific flows based on the configuration, for example, TCP Proxy can be enabled for some specific ports. TCP Proxy must be enabled at rulebase level. When enabled in a rulebase, it is applied on subscribers’ flows using that rulebase.

TCP Proxy can be enabled in static or dynamic modes. In static mode TCP proxy is enabled for all server ports/flows for a rulebase. In the dynamic mode/Socket Migration TCP Proxy is enabled dynamically based on specified conditions. In case TCP proxy is started dynamically on a flow, the original client (MS) first starts the TCP connection with the final server. ECS keeps on monitoring the connection. Based on any rule-match/charging-action, it may happen that the connection will be proxied automatically. This activity is transparent to original client and original server. After dynamically enabling the proxy, ECS acts as TCP endpoint exactly in the same way it is when connection is statically proxied.

The functional/charging behavior of ECS for that particular connection before the dynamic proxy is started is exactly same as when there is no proxy. After the dynamic proxy is started on the connection, the functional/charging behavior of the ECS for that particular connection will be exactly similar to the ECS static proxy behavior. When the socket migration is underway, the functional/charging behavior for that particular connection is exactly the same as when there is no proxy for that flow.

TCP Proxy impacts post-recovery behavior and the charging model. With TCP Proxy, whatever packets are received from either side is charged completely. The packets that are sent out from the ECS are not considered for charging. This approach is similar to the behavior of ECS without proxy.
The following packets will be charged at ECS:

- Uplink packets received at Gn interface
- Downlink packets received at Gi interface

The following packets will not be considered for charging:

- Uplink packets forwarded/sent out by ECS/Stack on the Gi interface.
- Downlink packets forwarded/sent out by ECS/Stack on the Gn interface.

ECS supports bulkstats for the TCP Proxy feature. For details see the ECS Schema Statistics chapter of the Statistics and Counters Reference.

**Flow Admission Control**

The Flow Admission Control feature controls the number of flows required to be proxied. It restricts admission of new calls based on the current resource usage, thus preventing system hog and service degradation to existing subscribers.

The number of flows required to be proxied will greatly depend on the deployment scenario. Operators have the provision to configure an upper bound on the memory used by proxy flows. This is specified as a percentage of the Session Manager memory that may be used for proxy flows. When memory utilization by existing proxy flows reaches this value, no further flows will be proxied.

Operators can also set a limit on the number of flows that can be proxied per subscriber. This would exercise Fair Usage policy to a certain extent. No credit usage information by proxy is communicated to the Session Manager.

**TCP Proxy Behavior and Limitations**

The following are behavioral changes applicable to various ECS features and on other applications after enabling TCP Proxy.

- **TCP Proxy Model:** Without TCP Proxy, for a particular flow, there is only a single TCP connection between subscriber and server. ECS is a passive entity with respect to flows and the packets received on ingress were sent out on egress side (except in case where some specific actions like drop are configured through CLI) transparently.

  With TCP Proxy, a flow is split into two TCP connections — one between subscriber and proxy and another between chassis and server.

- **Ingress Data Flow to Proxy:** For all uplink packets, ingress flow involves completing the following steps and then enters the Gn side TCP IP Stack of proxy:
  1. IP Analysis (support for IP reassembly)
  2. Shallow/Deep Packet TCP Analysis (support for TCP OOO)
  3. Stateful Firewall Processing
  4. Application Detection and Control Processing
  5. DPI Analysis
  6. Charging Function (including rule-matching, generation of various records, and applying various configured actions)

For all downlink packets, ingress flow would involve completing the following steps, and then enters the Gi side TCP IP Stack of proxy:

  1. IP Analysis (support for IP reassembly)
  2. Network Address Translation Processing
3. Shallow/Deep Packet TCP Analysis (support for TCP OOO)
4. Stateful Firewall Processing
5. Application Detection and Control Processing
6. DPI Analysis
7. Charging Function (including rule-matching, generation of various records, and applying various configured actions)

- Egress Data Flow from Proxy: All egress data flow is generated at proxy stack. For uplink packets, egress data flow would involve the following and then are sent out of the chassis:
  1. IP Analysis
  2. Shallow/Deep Packet TCP Analysis
  3. Stateful Firewall processing
  4. Network Address Translation processing

For downlink packets, egress data flow would involve the following and then are sent out of the chassis:
  1. IP Analysis
  2. Shallow/Deep Packet TCP Analysis
  3. Stateful Firewall processing

On enabling TCP Proxy the behavior of some ECS features will get affected. For flows on which TCP Proxy is enabled it is not necessary that all the packets going out of the Gn (or Gi) interface are the same (in terms of number, size, and order) as were on Gi (or Gn).

- IP Reassembly: If the fragments are successfully reassembled then DPI analysis is done on the reassembled packet.
  
  Without TCP Proxy, fragmented packets will go out on the other side. With TCP proxy, normal (non-fragmented) IP packets will go out on the other side (which will not be similar to the incoming fragmented packets).
  
  With or without TCP Proxy, if fragment reassembly was not successful, then all the fragments will be dropped except under the case where received fragments were sufficient enough to identify the 5-tuple TCP flow and the flow had TCP Proxy disabled on it.

- TCP OOO Processing: Without TCP Proxy if it is configured to send the TCP OOO packets out (as they come), without TCP proxy such packets were sent out. With TCP Proxy, OOO packets coming from one side will go in-order on the other side. For proxied flows TCP OOO expiry timer will not be started and hence there will be no specific handling based on any such timeouts. Also, TCP OOO packets will not be sent to other side unless the packets are re-ordered.

In releases prior to 14.0, when TCP Out-of-Order (OOO) packets were received and when there was any error in buffering those packets at ECS due to memory allocation failure, these packets were marked as TCP error packets and the rule matching was done accordingly. These packets were also marked as TCP error packets when the reordering packet was not received before the OOO timeout.

In 14.0 and later releases, in the above mentioned scenarios, the packets are not considered as TCP error and the TCP error flag is not set for OOO packets. So, these packets will not match TCP error related ruledef but match other appropriate ruledefs.

If the customer has configured TCP error related rules, then OOO timeout failure packets and memory allocation failure packets will not match these rules now. It will match normal TCP rules.

- TCP Checksum Validation: Without TCP Proxy TCP Checksum validation is optional (configurable through "transport-layer-checksum verify-during-packet-inspection tcp" CLI command). With TCP Proxy TCP
checksum is automatically done irrespective of whether the CLI command is configured or not. If the checksum validation fails, the packet is not processed further and so it does not go for application layer analysis.

- TCP Reset Packet Validation: Without TCP Proxy TCP reset packet is not validated for Seq and ACK number present in the segment and the flow is cleared immediately.

  With TCP Proxy TCP Reset packet validation is done. The flow will be cleared only if a valid TCP Reset segment is arrived. This validation is not configurable.

- TCP Timestamp (PAWS) Validation: Without TCP Proxy timestamp verification is not performed and even if there is any timestamp error, the packet is processed normally and goes for further analysis and rule-matching.

  With TCP Proxy if the connection is in established state, timestamp validation for packets is performed. If TCP timestamp is less than the previous timestamp, the packet is marked TCP error packet and is dropped. The packet is not analyzed further and not forwarded ahead. This packet should match TCP error rule (if configured). This validation is not configurable.

- TCP Error Packets: Without TCP Proxy ECS being a passive entity, most of the errors (unless configured otherwise) were ignored while parsing packets at TCP analyzer and were allowed to pass through. With TCP Proxy TCP error packets are dropped by Gi and Gn side TCP IP stack. However, since the ECS processing is already done before giving the packet to the stack, these packets are charged but not sent out by proxy on the other end.

- Policy Server Interaction (Gx): With TCP Proxy, application of policy function occurs on two separate TCP connections (non-proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and Internet) will be subject to policy enforcement at the box. This does not have any functional impact.

- Credit Control Interaction (Gy): With TCP Proxy, application of Credit Control function occur on two separate TCP connections (non-proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and Internet) will be subject to credit control at the box. This does not have any functional impact.

- DPI Analyzer: With TCP Proxy, application of DPI analyzer occurs on two separate TCP connections (non-proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and Internet) will be subject to DPI analyzer at the chassis. Any passive analyzer in the path would be buffering packet using the existing ECS infrastructure.

- ITC/BW Control: With TCP Proxy, only incoming traffic is dropped based on bandwidth calculation on ingress side packets. The BW calculation and dropping of packet is be done before sending packet to ingress TCP IP Stack. ToS and DSCP marking will be on flow level. The ToS and DSCP marking can be done only once for whole flow and once the ToS is marked for any packet either due to "ip tos" CLI command configured in the charging action or due to ITC/BW control, it will remain same for the whole flow.

- Next Hop and VLAN-ID: Without TCP Proxy nexthop feature is supported per packet, that is nexthop address can be changed for each and every packet of the flow depending on the configuration in the charging action. With TCP Proxy only flow-level next-hop will be supported. So, once the nexthop address is changed for any packet of the flow, it will remain same for the complete flow. The same is the case for VLAN-ID.

- TCP state based rules: Without TCP Proxy there is only one TCP connection for a flow and the TCP state based rules match to state of subscriber stack. With TCP Proxy there are two separate connections when TCP proxy is enabled. TCP state ("tcp state" and "tcp previous-state") based rules will match to MS state on egress side. Two new rules (tcp proxy-state and tcp proxy-prev-state) have been added to support the existing cases (of TCP state based rules). "tcp proxy-state" and "tcp proxy-prev-state" are the state of the embedded proxy server, that is the proxy ingress-side. These rules will not be applicable if proxy is not enabled.

  Using both "tcp state" and "tcp proxy-state" in the same ruledef is allowed. If proxy is enabled, they would map to Gi-side and Gn-side, respectively. If TCP Proxy is not enabled, the "tcp proxy-state" and "tcp proxy-prev-state" rules will not be matched because proxy-state will not be applicable.
Since TCP state and previous-state rules are now matched based on state on Gi side connection, ECS will not be able to support all the existing use-cases with the existing configuration. New ruledefs based on the new rules (tcp proxy-state and tcp proxy-prev-state) need to be configured to support existing use cases. Note that even by configuring using new rules; all use-cases may not be supported. For example, detection of transition from TIME-WAIT to CLOSED state is not possible now.

- TCP MSS: TCP IP Stack always inserts MSS Field in the header. This causes difference in MSS insertion behavior with and without TCP Proxy.
  - TCP CFG MSS limit-if-present: If incoming SYN has MSS option present, in outgoing SYN and SYN-ACK MSS value is limited to configured MSS value (CFG MSS)
  - TCP CFG MSS add-if-not-present: If incoming SYN does not have MSS option present, in outgoing SYN and SYN-ACK MSS configured MSS value is inserted (CFG MSS)
  - TCP CFG MSS limit-if-present add-if-not-present: If incoming SYN has MSS option present, in outgoing SYN and SYN-ACK MSS value is limited to configured MSS value (CFG MSS), OR if incoming SYN does not have MSS option present, in outgoing SYN and SYN-ACK MSS configured MSS value is inserted (CFG MSS).

- Flow Discard: Flow discard occurring on ingress/egress path of TCP Proxy would be relying on TCP-based retransmissions. Any discard by payload domain applications would result in data integrity issues as this might be charged already and it may not be possible to exclude packet. So it is recommended that applications in payload domain (like dynamic CF, CAE readdressing) should not be configured to drop packets. For example, dynamic content filtering should not be configured with drop action. If drop is absolutely necessary, it is better to use terminate action.

- DSCP/IP TOS Marking: Without TCP Proxy DSCP/IP TOS marking is supported per packet, that is IP TOS can be changed for each and every packet of the flow separately based on the configuration. With TCP Proxy flow-level DSCP/IP TOS marking is supported. So, once the IP TOS value is changed for any packet of the flow, it will remain same for the complete flow.

- Redundancy Support (Session Recovery and ICSR): Without TCP Proxy after recovery, non-syn flows are not reset. With TCP Proxy session recovery checkpointing is bypassing any proxied flows (currently on NAT flows support recovery of flows). If any flow is proxied for a subscriber, after recovery (session recovery or ICSR), if any non-syn packet is received for that subscriber, ECS sends a RESET to the sender. So, all the old flows will be RESET after recovery.

- Charging Function: Application of charging function would occur on two separate TCP connections (non proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and Internet) shall be subject to Policy enforcement at the box. Offline charging records generated at charging function would pertain to different connections hence.

**Dynamic Disabling of TCP Proxy**

TCP proxy can be dynamically disabled to reduce the performance overhead on CPU and memory resources. This enables applications to use proxy only when required.

Dynamic disabling is achieved by merging the TCP connections. Before dynamic disabling occurs, the packets are added to a TCP stack with a full proxy connection. Once proxy is disabled dynamically, the TCP stack and proxy are removed from the data processing path and the packets are forwarded without buffering.

Disabling of TCP proxy dynamically occurs only after the following conditions are met:

- There is no data to be delivered by ECS to the peer.
- The flow control buffers do not contain any data.
- There is no data to be read by ECS.
Limitations for Dynamically Disabling TCP Proxy

This section lists known limitations to disabling TCP proxy dynamically:

- TCP proxy cannot be disabled when one end of the TCP supports time stamp and other does not.
- Dynamic disabling does not work when both sides of the TCP have different MSS negotiated.
- Toggling the proxy on the same connection might reduce TCP performance.
- TCP proxy can only be disabled when both ends of TCP are in connected states.
- Multiple connections (1:n) connections cannot be joined together.
- TCP proxy can only be disabled when the conditions outlined for dynamic disabling is achieved (when there is no unAcked data in the network).

X-Header Insertion and Encryption

This section describes the X-Header Insertion and Encryption features, collectively known as Header Enrichment, which enable to append headers to HTTP/WSP GET and POST request packets for use by end applications, such as mobile advertisement insertion (MSISDN, IMSI, IP address, user-customizable, and so on).

Important: In this release, the X-Header Insertion and Encryption features are supported only on the GGSN, IPSG, and P-GW.

License Requirements

X-Header Insertion and Encryption are both licensed Cisco features. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the Managing License Keys section of the Software Management Operations chapter in the System Administration Guide.

X-Header Insertion

This section provides an overview of the X-Header Insertion feature.

Extension header (x-header) fields are the fields not defined in RFCs or standards but can be added to headers of protocol for specific purposes. The x-header mechanism allows additional entity-header fields to be defined without changing the protocol, but these fields cannot be assumed to be recognizable by the recipient. Unrecognized header fields should be ignored by the recipient and must be forwarded by transparent proxies.

The X-Header Insertion feature enables inserting x-Headers in HTTP/WSP GET and POST request packets. Operators wanting to insert x-Headers in HTTP/WSP GET and POST request packets, can configure rules for it. The charging-action associated with the rules will contain the list of x-Headers to be inserted in the packets.

For example, if you want to insert the field x-rat-type in the HTTP header with a value of rat-type, the header inserted should be:

x-rat-type: geran

where, rat-type is geran for the current packet.

Configuring the X-Header Insertion feature involves:
**Enhanced Charging Service Overview**

**Enhanced Features and Functionality**

- **Cisco ASR 5x00 Enhanced Charging Services Administration Guide**

---

**Step 1** Creating/configuring a ruledef to identify the HTTP/WSP packets in which the x-headers must be inserted.

**Step 2** Creating/configuring a rulebase and configuring the charging-action, which will insert the x-header fields into the HTTP/WSP packets.

**Step 3** Creating/configuring the x-header format.

**Step 4** Configuring insertion of the x-header fields in the charging action.

**X-Header Encryption**

This section provides an overview of the X-Header Encryption feature.

X-Header Encryption enhances the X-header Insertion feature to increase the number of fields that can be inserted, and also enables encrypting the fields before inserting them.

If x-header insertion has already happened for an IP flow (because of any x-header format), and if the current charging-action has the first-request-only flag set, x-header insertion will not happen for that format. If the first-request-only flag is not set in a charging-action, then for that x-header format, insertion will continue happening in any further suitable packets in that IP flow.

Changes to x-header format configuration will not trigger re-encryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next re-encryption time to those existing calls for which re-encryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value will not be available, insertion of that parameter will stop.

---

**Important:** Recovery of flows is not supported for this feature.

The following steps describe how X-Header Encryption works:

- **Step 1** X-header insertion, encryption, and the encryption certificate is configured in the CLI.
- **Step 2** When the call gets connected, and after each regeneration time, the encryption certificate is used to encrypt the strings.
- **Step 3** When a packet hits a ruledef that has x-header format configured in its charging-action, x-header insertion into that packet is done using the given x-header-format.
- **Step 4** If x-header-insertion is to be done for fields which are marked as encrypt, the previously encrypted value is populated for that field accordingly.

**TCP OOO Packets**

ECS handles TCP OOO packets in two ways depending on the rulebase configuration:

**Transmit Immediately:** If the rulebase is configured to transmit immediately for TCP OOO packets, the OOO packets will be forwarded immediately, and a copy of this packet will be added to the OOO queue for analysis.

**Transmit After Reordering:** If the rulebase is configured to transmit after reordering for TCP OOO packets, the OOO packets will be added to the OOO queue for analysis. Header insertion on OOO request packets occurs on reordering packets that are received before the OOO request timeout. When a reordering packet is received, the queued packets are forwarded. However, if a reordering packet is not received before the OOO queue timeout, the queued packet will be forwarded without any analysis done to those packets.

**IP Fragmented Packets**
ECS can perform Header Enrichment to IP fragmented packets when all the fragments are received before the reassembly timeout. If the packet size after Header Enrichment exceeds the MSS of the session, the reassembled packet gets segmented, the multiple segments are forwarded.

**Limitations to the Header Insertion and Encryption Features**

This section lists known limitations to insertion and encryption of x-header fields in HTTP and WSP headers.

The following are limitations to insertion and encryption of x-header fields in HTTP headers.

**Limitations in StarOS 14.0 and later releases:**

- Header insertion does not occur for packets with incomplete HTTP headers.
- If a flow has x-header insertion and later some IP fragments are received for which reassembly fails, sequence space of that segment will be mismatched.
- ECS does not support applying more than one modifying action on an inbound packet before sending it on the outbound interface. For example, if header insertion is applied on a packet, then the same packet is not allowed to be modified for NAT/ALG and MSS insertion.
- Header enrichment works only for the first request of a packet with concatenated requests, when the packets are buffered at DCCA. There are no limitations on header enrichment for single GET or pipelined GET requests.
- Header enrichment works for packets at DCCA only when the packets pending of header insertion is buffered.
- Receive window will not be considered during header enrichment. That is, after header enrichment if packet exceeds receive window, ECS will not truncate the packet.
- The maximum bytes per request after header enrichment is 2400 bytes. If concatenated requests exist, a maximum of 2400 bytes after header enrichment can be inserted.
  - If due to header insertion, the packet size exceeds this limit, the behavior is unpredictable.
- Only those x-header fields in header portion of application protocol that begin with “x-” are parsed at HTTP analyzer. In URL and data portion of HTTP any field can be parsed.

**Limitations in StarOS 12.3 and earlier releases:**

- The packet size is assumed to be less than “Internal MED MTU size, the size of header fields inserted”. If the total length of packet exceeds the internal MTU size, header insertion will not occur after the addition of fields.
- Header insertion occurs for both HTTP GET and POST requests. However, for POST requests, the resulting packet size will likely be larger than for GET requests due to the message body contained in the request. If the previous limitation applies, then POST request will suffer a bigger limit due to this.
- Header insertion does not occur for retransmitted packets.
- Header insertion does not occur for packets with incomplete HTTP headers.
- Header insertion does not occur for TCP OOO and IP fragmented packets.
- If a flow has x-header insertion and later some IP fragments are received for which reassembly fails, sequence space of that segment will be mismatched.
- ECS does not support applying more than one modifying action on an inbound packet before sending it on the outbound interface. For example, if header insertion is applied on a packet, then the same packet is not allowed to be modified for NAT/ALG and MSS insertion.
- If a packet is buffered by ICAP, header insertion will not occur for that packet.
- Receive window will not be considered during header enrichment. That is, after header enrichment if packet exceeds receive window, ECS will not truncate the packet.
• Packet size limit is 2400 bytes, if due to header insertion packet size exceeds this limit, behavior is unpredictable.

• Only those x-header fields in header portion of application protocol that begin with “x-” are parsed at HTTP analyzer. In URL and data portion of HTTP any field can be parsed.

The following are limitations to insertion and encryption of x-header fields in WSP headers:

• x-header fields are not inserted in IP fragmented packets.

• In case of concatenated request, x-header fields are only inserted in first GET or POST request (if rule matches for the same). X-header fields are not inserted in the second or later GET/POST requests in the concatenated requests. For example, if there is ACK+GET in packet, x-header is inserted in the GET packet. However, if GET1+GET2 is present in the packet and rule matches for GET2 and not GET1 x-header is still inserted in GET2. In case of GET+POST also, x-header is not inserted in POST.

• In case of CO, x-header fields are not inserted if the WTP packets are received out of order (even after proper re-ordering).

• If route to MMS is present, x-headers are not inserted.

• x-headers are not inserted in WSP POST packet when header is segmented. This is because POST contains header length field which needs to be modified after addition of x-headers. In segmented WSP headers, header length field may be present in one packet and header may complete in another packet.

• x-headers are not inserted in case of packets buffered at DCCA.

Supported X-Headers

This section provides information on the different x-headers supported by ECS.

ECS supports insertion of various x-header fields in the HTTP/WSP GET and POST request packets. The x-headers are inserted at the end of the HTTP/WSP header.

The following bearer-related x-headers are supported:

• 3gpp

  The following 3GPP associated fields are supported:

  • apn
  • charging-characteristics
  • charging-id
  • imei
  • imsi
  • qos
  • rat-type
  • s-mcc-mnc
  • sgsn-address

  • acr
  • customer-id
  • ggsn-address
  • mdn
The following HTTP-related x-headers are supported:

- host
- url

In addition, ECS also allows string constants to be inserted as an x-header. For more information on configuring the x-header formats, see the insert command section in the ACS x-Header Format Configuration Mode Commands chapter of the Command Line Interface Reference.

**X-Header Enrichment Anti Spoofing**

This section provides an overview of the x-Header Enrichment Anti Spoofing feature. The Header Enrichment feature allows operators to encrypt and insert subscriber-specific fields as x-headers in to the HTTP headers of URL requests. However, this might leave the header open to spoofing by malicious external devices. The Anti Spoofing feature enables deletion and modification of the existing x-header fields to protect the operators and subscribers from spoofing, and provides a fraud detection mechanism when an external portal is used for a subscriber or content authorization.

The feature detects and removes user-generated HTTP headers if the header name is similar to the header name used in the x-header format, and when multiple entries of the same field exist in the header, all the similar entries are removed and one with a modified value is inserted at the end of the HTTP header.

When anti spoofing is enabled, and if the HTTP header in the GET or POST request spawns across more than one packet, the packets with incomplete HTTP header will be buffered. The buffered packets are sent out once the HTTP header is completed.

The Anti Spoofing feature is disabled by default and can be enabled/disabled at a field level in the CLI.

**Limitations to the Anti-Spoofing Feature:**

- Header enrichment does not occur if a route to the MMS analyzer exist in the rulebase.
- Header enrichment works only for the first request of a packet with concatenated requests, when the packets are buffered at DCCA.
- If a packet is buffered by ICAP, header insertion will not occur for that packet.
- ECS will not be able to perform header enrichment when all fragments are not received before reassembly timeout in the case of IP fragments packets.
- ECS does not perform more than one flow action which modifies the inbound packet before sending it on the outbound interface.
- If the HTTP GET or POST header is not completed in three packets, anti spoofing will occur only for the last packet in which the header completes, as buffering supported only up to a maximum of two packets.
RAN Bandwidth Optimization

When the rule is installed and active, P-GW uses the GBR/MBR assigned in the rule for calculating the GBR / MBR values towards the bearers created. When more than one rule is installed, P-GW adds the GBR / MBR values from all the active and installed rules even if the flow of a certain rule is marked as disabled. This current behavior is in accordance with 3GPP TS standard specification 29.212, and this might result in RAN bandwidth wastage. To avoid this wastage, some optimization is done while calculating MBR and GBR for GBR bearer.

The RAN bandwidth optimization feature provides the ability to configure a list of APNs, for which the optimized calculation of MBR, GBR can be enabled. By default, this optimized calculation should be enabled only for the IMS APN.

This feature further helps optimize the logic of aggregating MBR and GBR values, based on “Flow-Status” AVP value received in the rule definition through RAR. This operation is controlled through a CLI command “ran bandwidth optimize” added in the rulebase configuration mode.

For more information on this command, see the ACS Rulebase Configuration Mode Commands chapter in the Command Line Interface Reference.

Selective TFT Suppression for Default Bearer

With this feature, the selected TFT updates can be controlled and sent to the UE. A new CLI command “tft-notify-ue” is introduced, which suppresses the selected TFT updates to the UE. This is provided by specific charging-action level option to identify if the appropriate TFT defined in the charging action needs to be sent to the UE or not. This CLI is supported for both default and dedicated bearer.

One more new CLI “tft-notify-ue-def-bearer” to suppress TFTs on default bearer has been added, so the operator has the flexibility to configure this per Rulebase and also configure to suppress TFT updates only. This CLI allows sending other QoS updates to the UE and is only controlling TFT related updates. This CLI is supported only for default bearer.

For more information on these commands, see the ACS Charging Action Configuration Mode Commands and ACS Rulebase Configuration Mode Commands chapters in the Command Line Interface Reference.
Accounting and Charging Interfaces

ECS supports different accounting and charging interfaces for prepaid and postpaid charging and record generation.

**Important:** Some feature described in this section are licensed Cisco features. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the Managing License Keys section of the Software Management Operations chapter in the System Administration Guide.

Accounting Interfaces for Postpaid Service: ECS supports the following accounting interfaces for postpaid subscribers:
- Remote Authentication Dial-In User Service (RADIUS) Interface
- GTPP Accounting Interface

Accounting and Charging Interface for Prepaid Service: ECS supports the following Credit Control Interfaces for prepaid subscribers:
- RADIUS Prepaid Credit Control interface
- Diameter Prepaid Credit Control Application (DCCA) Gy Interface
- Diameter Gx interface

Charging Records in ECS: ECS provides the following charging records for postpaid and prepaid charging:
- GGSN-Call Detail Records (G-CDRs)
- Enhanced GGSN-Call Detail Records (eG-CDRs)
- Event Detail Records (EDRs)
- Usage Detail Records (UDRs)

**GTPP Accounting**

ECS enables the collection of counters for different types of data traffic, and including that data in CDRs that is sent to a Charging Gateway Function (CGF).

For more information on GTPP accounting, if you are using StarOS 12.3 or an earlier release, refer to the AAA and GTPP Interface Administration and Reference. If you are using StarOS 14.0 or a later release, refer to the GTPP Interface Administration and Reference.

**RADIUS Accounting and Credit Control**

The Remote Authentication Dial-In User Service (RADIUS) interface in ECS is used for the following purposes:
- **Subscriber Category Request**—ECS obtains the subscriber category from the AAA server (either prepaid or postpaid) when a new data session is detected. The AAA server used for the subscriber category request can be different from the AAA server used for service authorization and accounting.
- **Service Access Authorization**—ECS requests access authorization for a specific subscriber and a newly detected data session. The AAA server is the access Policy Decision Point and the ECS the Policy Enforcement Point.
• **On-line Service Accounting (Prepaid)**—ECS reports service usage to the AAA server. The AAA server acts as a prepaid control point and the ECS as the client. Accounting can be applied to a full prepaid implementation or just to keep ECS updated of the balance level and trigger a redirection if the subscriber balance reaches a low level.

### Diameter Accounting and Credit Control

The Diameter Credit Control Application (DCCA) is used to implement real-time online or offline charging and credit control for a variety of services, such as network access, messaging services, and download services.

In addition to being a general solution for real-time cost and credit control, DCCA includes these features:

- **Real-time Rate Service Information**—DCCA can verify when end subscribers' accounts are exhausted or expired; or deny additional chargeable events.

- **Support for Multiple Services**: DCCA supports the usage of multiple services within one subscriber session. Multiple service support includes:
  - The ability to identify and process the service or group of services that are subject to different cost structures.
  - Independent credit control of multiple services in a single credit control sub-session.

### Gx Interface Support

The Gx interface is used in IMS deployment in GPRS/UMTS networks. Gx interface support on the system enables wireless operators to intelligently charge the services accessed depending on the service type and parameters with rules. It also provides support for IP Multimedia Subsystem (IMS) authorization in a GGSN service. The goal of the Gx interface is to provide network-based QoS control as well as dynamic charging rules on a per bearer basis for an individual subscriber. The Gx interface is in particular needed to control and charge multimedia applications.

**Important:** For more information on Gx interface support, see the *Gx Interface Support* appendix in the administration guide for the product that you are deploying.

### Gy Interface Support

The Gy interface provides a standardized Diameter interface for real-time content-based charging of data services. It is based on the 3GPP standards and relies on quota allocation.

It provides an online charging interface that works with the ECS Deep Packet Inspection feature. With Gy, customer traffic can be gated and billed in an “online” or “prepaid” style. Both time- and volume-based charging models are supported. In all these models, differentiated rates can be applied to different services based on shallow or deep-packet inspection.

Gy is a Diameter interface. As such, it is implemented atop, and inherits features from, the Diameter Base Protocol. The system supports the applicable base network and application features, including directly connected, relayed or proxied DCCA servers using TLS or plain text TCP.

In the simplest possible installation, the system will exchange Gy Diameter messages over Diameter TCP links between itself and one “prepay” server. For a more robust installation, multiple servers would be used. These servers may optionally share or mirror a single quota database so as to support Gy session failover from one server to the other. For a more scalable installation, a layer of proxies or other Diameter agents can be introduced to provide features such as multi-path message routing or message and session redirection features.
The Diameter Credit Control Application (DCCA) which resides as part of the ECS manages the credit and quota for a subscriber.

**Important:** For more information on Gy interface support, see the Gy Interface Support appendix in the administration guide for the product that you are deploying.

### Event Detail Records (EDRs)

Event Detail Records (EDRs) are usage records with support to configure content information, format, and generation triggers by the system administrative user.

EDRs are generated according to explicit action statements in rule commands. Several different EDR schema types, each composed of a series of analyzer parameter names, are specified in EDR. EDRs are written at the time of each event in CSV format. EDRs are stored in timestamped files that can be downloaded via SFTP from the configured context.

EDRs are generated on per flow basis, and as such they catch whatever bytes get transmitted over that flow including retransmitted.

### EDR format

The EDRs can be generated in comma separated values (CSV) format as defined in the traffic analysis rules.

**Important:** In EDRs, the maximum field length for normal and escaped strings is 127 characters. If a field’s value is greater than 127 characters, in the EDR it is truncated to 127 characters. In 15.0 and later releases, an optional filter “length” is supported for HTTP URL and User-Agent fields which when added will allow the user to configure length from 1 to 255 for these fields in EDRs. For more information, see the rule-variable command in the ASR 5x00 Command Line Interface Reference.

### Flow-overflow EDR

Flow-overflow EDR or Summary FDR is a feature to count the data bytes from the subscriber that are missed due to various reasons in ECS.

In case any condition that affects the call line (FLOW end-condition like hagr, handoff) occurs, flow-overflow EDR generation is enabled, an extra EDR is generated. Based on how many bytes/packets were transferred from/to the subscriber for which ECS did not allocate data session. This byte/packet count is reflected in that extra EDR. This extra EDR is nothing but “flow-overflow” EDR or Summary FDR.

The extra EDR is generated if all of the following is true:

- Subscriber affecting condition occurs (session-end, hand-off, hagr)
- Flow-overflow EDR generation is enabled
- EDR generation on session-end, hand-off or hagr is enabled
- Number of bytes/packets for flow-overflow EDR is non-zero.

The bytes/packet count will be printed as a part of “sn-volume-amt” attribute in the EDR. Hence, this attribute must be configured in the EDR format.

### EDR Generation in Flow-end and Transaction Complete Scenarios with sn-volume Fields
“sn-volume-amt” counters will be re-initialized only when the fields are populated in EDRs. For example, consider the following two EDR formats:

```plaintext
edr-format edr1
   rule-variable http url priority 10
   attribute sn-volume-amt ip bytes uplink priority 500
   attribute sn-volume-amt ip bytes downlink priority 510
   attribute sn-volume-amt ip pkts uplink priority 520
   attribute sn-volume-amt ip pkts downlink priority 530
   attribute sn-app-protocol priority 1000
   exit

edr-format edr2
   rule-variable http url priority 10
   attribute sn-app-protocol priority 1000
   exit
```

“sn-volume-amt counters” will be re-initialized only if these fields are populated in the EDRs. Now if edr2 is generated, these counters will not be re-initialized. These will be re-initialized only when edr1 is generated. Also, note that only those counters will be re-initialized which are populated in EDR. For example, in the following EDR format:

```plaintext
edr-format edr3
   rule-variable http url priority 10
   attribute sn-volume-amt ip bytes uplink priority 500
   attribute sn-volume-amt ip bytes downlink priority 510
   attribute sn-app-protocol priority 1000
   exit
```

If edr3 is generated, only uplink bytes and downlink bytes counter will be re-initialized and uplink packets and downlink packets will contain the previous values till these fields are populated (say when edr1 is generated).

For the voice call duration for SIP reporting requirements, ECS SIP analyzer keeps timestamp of the first INVITE that it sees. It also keeps a timestamp when it sees a 200 OK for a BYE. When this 200 OK for a BYE is seen, SIP analyzer triggers creation of an EDR of type ACS_EDR_VOIP_CALL_END_EVENT. This will also be triggered at the time of SIP flow termination if no 200 OK for BYE is seen. In that case, the last packet time will be used in place of the 200 OK BYE timestamp. The EDR generation logic calculates the call duration based on the INVITE and end timestamps, it also accesses the child RTP/RTCP flows to calculate the combined uplink/downlink bytes/packets counts and sets them in the appropriate fields.
Usage Detail Records (UDRs)

Usage Detail Records (UDRs) contain accounting information based on usage of service by a specific mobile subscriber. UDRs are generated based on the content-id for the subscriber, which is part of charging action. The fields required as part of usage data records are configurable and stored in the System Configuration Task (SCT).

UDRs are generated on any trigger of time threshold, volume threshold, handoffs, and call termination. If any of the events occur then the UDR subsystem generates UDRs for each content ID and sends to the CDR module for storage.

UDR format

The UDRs are generated in Comma Separated Values (CSV) format as defined in the traffic analysis rules.

Charging Methods and Interfaces

Prepaid Credit Control

Prepaid billing operates on a per content-type basis. Individual content-types are marked for prepaid treatment. A match on a traffic analysis rule that has a prepaid-type content triggers prepaid charging management.

In prepaid charging, ECS performs the metering function. Credits are deducted in real time from an account balance or quota. A fixed quota is reserved from the account balance and given to the system by a prepaid rating and charging server, which interfaces with an external billing system platform. The system deducts volume from the quota according to the traffic analysis rules. When the subscriber’s quota gets to the threshold level specified by the prepaid rating and charging server, system sends a new access request message to the server and server updates the subscriber’s quota. The charging server is also updated at the end of the call.

ECS supports the following credit control applications for prepaid charging:

- **RADIUS Credit Control Application**—RADIUS is used as the interface between ECS and the prepaid charging server. The RADIUS Prepaid feature of ECS is separate to the system-level Prepaid Billing Support and that is covered under a different license key.

- **Diameter Credit Control Application**—The Diameter Credit Control Application (DCCA) is used to implement real-time credit control for a variety of services, such as networks access, messaging services, and download services.

In addition to being a general solution for real-time cost and credit control, DCCA includes the following features:

- **Real-time Rate Service Information**—DCCA can verify when end subscribers' accounts are exhausted or expired; or deny additional chargeable events.

- **Support for Multiple Services**—DCCA supports the usage of multiple services within one subscriber session. Multiple service support includes:
  - The ability to identify and process the service or group of services that are subject to different cost structures.
  - Independent credit control of multiple services in a single credit control sub-session.

Postpaid

In a postpaid environment, the subscribers pay after use of the service. AAA/RADIUS server is responsible for authorizing network nodes to grant access to the user, and the CDR system generates G-CDRs/eG-CDRs/EDRs/UDRs for billing information on pre-defined intervals of volume or per time.
Important: G-CDRs and eG-CDRs are only available in UMTS networks.

ECS also supports FBC and TBC methods for postpaid billing. For more information on FBC and TBC in ECS, see the Time and Flow-based Bearer Charging in ECS section.

Prepaid Billing in ECS

In a prepaid environment, the subscribers pay for service prior to use. While the subscriber is using the service, credit is deducted from subscriber’s account until it is exhausted or call ends. The prepaid charging server is responsible for authorizing network nodes to grant access to the user, as well as grant quotas for either time connected or volume used. It is up to the network node to track the quota use, and when these use quotas run low, the network node sends a request to the prepaid server for more quota.

If the user has not used up the purchased credit, the server grants quota and if no credit is available to the subscriber the call will be disconnected. ECS and DCCA manage this functionality by providing the ability to set up quotas for different services.

Prepaid quota in ECS is implemented using RADIUS and DCCA as shown in the following figure.

How ECS Prepaid Billing Works

The following figure illustrates a typical prepaid billing environment with system running ECS.

Figure 9. Prepaid Billing Scenario with ECS
Credit Control Application (CCA) in ECS

This section describes the credit control application that is used to implement real-time credit-control for a variety of end user services such as network access, Session Initiation Protocol (SIP) services, messaging services, download services, and so on. It provides a general solution to the real-time cost and credit control.

CCA with RADIUS or Diameter interface uses a mechanism to allow the user to be informed of the charges to be levied for a requested service. In addition, there are services such as gaming and advertising that may debit from a user account.

How Credit Control Application (CCA) Works for Prepaid Billing

The following figure and steps describe how CCA works with in a GPRS/UMTS or CDMA-2000 network for prepaid billing.

Table 6. Prepaid Charging in GPRS/UMTS/CDMA-2000 Networks

<table>
<thead>
<tr>
<th>Step No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Subscriber session starts.</td>
</tr>
<tr>
<td>2</td>
<td>System sends request to CCA for subscriber’s quota.</td>
</tr>
<tr>
<td>Step No.</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>CCA sends request to Data Warehouse (DW) credit quota for subscriber.</td>
</tr>
<tr>
<td>4</td>
<td>Credit Database in DW sends pre-configured amount of usage limit from subscriber’s quota to CCA. To reduce the need for multiple requests during subscriber’s session configured amount of usage limit a major part of available credit quota for subscriber is set.</td>
</tr>
<tr>
<td>5</td>
<td>CCA sends the amount of quota required to fulfill the subscriber’s initial requirement to the system.</td>
</tr>
<tr>
<td>6</td>
<td>When the initial amount of quota runs out, system sends another request to the CCA and the CCA sends another portion of available credit quota.</td>
</tr>
<tr>
<td>7</td>
<td>Subscriber session ends after either quota exhausts for subscriber or subscriber terminates the session.</td>
</tr>
<tr>
<td>8</td>
<td>CCA returns unused quota to DW for update to subscribers Credit DB.</td>
</tr>
<tr>
<td>9</td>
<td>EDRs and UDRs are periodically SFTPd from system memory to the ESS/external storage, if deployed or to billing system directly as they are generated. Or, if configured, pushed to the ESS/external storage at user-configurable intervals.</td>
</tr>
<tr>
<td>10</td>
<td>The ESS/external storage periodically sends records to the billing system or charging reporting and analysis system.</td>
</tr>
</tbody>
</table>

**Important:** For information on ESS contact your Cisco account representative.

**Postpaid Billing in ECS**

This section describes the postpaid billing that is used to implement off-line billing processing for a variety of end user services.

The following figure shows a typical deployment of ECS for postpaid billing system.
How ECS Postpaid Billing Works

This section describes how the ECS postpaid billing works in the GPRS/UMTS and CDMA-2000 Networks.

ECS Postpaid Billing in GPRS/UMTS Networks

The following figure and steps describe how ECS works in a GPRS/UMTS network for postpaid billing.
Table 7. Postpaid Billing with ECS in GPRS/UMTS Network

<table>
<thead>
<tr>
<th>Step No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The subscriber initiates the session.</td>
</tr>
<tr>
<td>2</td>
<td>After subscriber authentication and authorization, the system starts the session.</td>
</tr>
<tr>
<td>3</td>
<td>Data packet flow and accounting starts.</td>
</tr>
<tr>
<td>4</td>
<td>System periodically generates xDRs and stores them to the system memory.</td>
</tr>
<tr>
<td>5</td>
<td>System generates G-CDRs/eG-CDRs and sends them to billing system as they are generated.</td>
</tr>
<tr>
<td>6</td>
<td>The billing system picks up the CDR files periodically.</td>
</tr>
<tr>
<td>7</td>
<td>Subscriber session ends after subscriber terminates the session.</td>
</tr>
<tr>
<td>8</td>
<td>The system stores the last of the xDRs to the system memory and final xDRs are SFTPd from system memory to ESS/external storage, if deployed or to billing system directly.</td>
</tr>
<tr>
<td>9</td>
<td>System sends the last of the G-CDRs/eG-CDRs to the billing system.</td>
</tr>
<tr>
<td>10</td>
<td>File Generation Utility, FileGen in external storage periodically runs to generate G-CDRs/eG-CDRs files for billing system and send them to the billing system.</td>
</tr>
<tr>
<td>11</td>
<td>The billing system picks up the xDR files from the ESS/external storage periodically.</td>
</tr>
</tbody>
</table>

Postpaid Billing in CDMA-2000 Networks

The following figure and steps describe how ECS works within a CDMA-2000 network for postpaid billing.

Figure 13. Postpaid Billing with ECS in CDMA-2000 Network
## Table 8. Postpaid Billing with ECS in GPRS/UMTS Network

<table>
<thead>
<tr>
<th>Step No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The subscriber initiates the session.</td>
</tr>
<tr>
<td>2</td>
<td>After subscriber authentication and authorization, the system starts the session.</td>
</tr>
<tr>
<td>3</td>
<td>Data packet flow and accounting starts.</td>
</tr>
<tr>
<td>4</td>
<td>System periodically generates xDRs and stores them to the system memory.</td>
</tr>
<tr>
<td>5</td>
<td>EDRs/UDRs are periodically SFTPd from system memory to ESS/external storage, if deployed or to billing system directly as they are generated.</td>
</tr>
<tr>
<td>6</td>
<td>The billing system picks up the xDR files from the ESS/external storage periodically.</td>
</tr>
<tr>
<td>7</td>
<td>Subscriber session ends after subscriber terminates the session.</td>
</tr>
<tr>
<td>8</td>
<td>The system stores the last of the xDRs to the system memory and final xDRs are SFTPd from system memory to the ESS/external storage, if deployed or to billing system directly.</td>
</tr>
<tr>
<td>9</td>
<td>The ESS/external storage finally sends xDRs to the billing system.</td>
</tr>
</tbody>
</table>
External Storage System

**Important:** For information on availability/support for ESS, contact your Cisco account representative.

The External Storage System (ESS) is a high availability, fault tolerant, redundant solution for short-term storage of files containing detail records (UDRs/EDRs/FDRs (xDRs)). To avoid loss of xDRs on the chassis due to overwriting, deletion, or unforeseen events such as power or network failure or unplanned chassis switchover, xDRs are off-loaded to ESS for storage and analysis to avoid loss of charging and network analysis information contained in the xDRs.

The xDR files can be pulled by the ESS from the chassis, or the chassis can push the xDR files to the ESS using SFTP protocol. In the Push mode, the ESS URL to which the CDR files need to be transferred to is specified. The configuration allows a primary and a secondary server to be configured. Configuring the secondary server is optional. Whenever a file transfer to the primary server fails for four consecutive times, the files will be transferred to the secondary server. The transfer will switch back to the original primary server when:

- Four consecutive transfer failures to the secondary server occur
- After switching from the primary server, 30 minutes elapses

In the push transfer mode, the following can be configured:

- Transfer interval—A time interval, in seconds, after which the CDRs are pushed to the configured IP periodically. All the files that are completed before the PUSH timer expires are pushed.
- Remove file after transfer—An option to keep or remove the CDR files on the hard disk after they are transferred to the ESS successfully.

The system running with ECS stores xDRs on an ESS, and the billing system collects the xDRs form the ESS and correlates them with the AAA accounting messages using 3GPP2-Correlation-IDs (for PDSN) or Charging IDs (for GGSN).

**Important:** For more information on the ESS, refer to the *ESS Installation and Administration Guide*. 
System Resource Allocation

ECS does not require manual resource allocation. The ECS subsystem automatically allocates the resources when ECS is enabled on the chassis. ECS must be enabled on the chassis before configuring services.
Redundancy Support in ECS

This section describes the redundancy support available in ECS to recover user sessions and charging records in the event of software/hardware failure.

⚠️ **Caution:** Persistent data flows are NOT recoverable during session recovery.

ℹ️ **Important:** Redundancy is not available in the current version of the Cisco XT2 platform.

### Intra-chassis Session Recovery Interoperability

Intra-chassis session recovery is coupled with SessMgr recovery procedures.

Intra-chassis session recovery support is achieved by mirroring the SessMgr and AAAMgr processes. The SessMgrs are paired one-to-one with the AAAMgrs. The SessMgr sends checkpointed session information to the AAAMgr. ECS recovery is accomplished using this checkpointed information.

ℹ️ **Important:** In order for session recovery to work there should be at least four packet processing cards, one standby and three active. Per active CPU with active SessMgrs, there is one standby SessMgr, and on the standby CPU, the same number of standby SessMgrs as the active SessMgrs in the active CPU.

There are two modes of session recovery, one from task failure and another on failure of CPU or packet processing card.

#### Recovery from Task Failure

When a SessMgr failure occurs, recovery is performed using the mirrored “standby-mode” SessMgr task running on the active packet processing card. The “standby-mode” task is renamed, made active, and is then populated using checkpointed session information from the AAAMgr task. A new “standby-mode” SessMgr is created.

#### Recovery from CPU or Packet Processing Card Failure

When a PSC, PSC2, or PPC hardware failure occurs, or when a planned packet processing card migration fails, the standby packet processing card is made active and the “standby-mode” SessMgr and AAAMgr tasks on the newly activated packet processing card perform session recovery.

### Inter-chassis Session Recovery Interoperability

The system supports the simultaneous use of ECS and the Inter-chassis Session Recovery feature. (For more information on the Inter-chassis Session Recovery feature, refer to the *System Administration Guide.*) When both features are enabled, ECS session information is regularly checkpointed from the active chassis to the standby as part of normal Service Redundancy Protocol processes.

In the event of a manual switchover, there is no loss of accounting information. All xDR data from the active chassis is moved to a customer-configured ESS before switching over to the standby. This data can be retrieved at a later time.
Upon completion of the switchover, the ECS sessions are maintained and the “now-active” chassis recreates all of the session state information including the generation of new xDRs. In the event of an unplanned switchover, all accounting data that has not been written to the external storage is lost. (Note that either the ESS can pull the xDR data from the chassis, or the chassis can push the xDR files to a configured ESS at user-configured intervals. For more information, see External Storage System section.) Upon completion of switchover, the ECS sessions are maintained and the “now-active” chassis recreates all of the session state information including the generation of new xDRs.

Regardless of the type of switchover that occurred, the names of the new xDR files will be different from those stored in the /records directory of packet processing card RAM on the “now-standby” chassis. Also, in addition to the file name, the content of many of the fields within the xDR files created by the “now-active” chassis will be different. ECS manages this impact with recovery mechanism. For more information on the differences and how to correlate the two files and other recovery information, see the Impact on xDR File Naming section.

Inter-chassis Session Recovery Architecture

Inter-chassis redundancy in ECS uses Flow Detail Records (FDRs) and UDRs to manage the switchover between Active-Standby system. xDRs are moved between redundant external storage server and Active-Standby systems.

Session Recovery Improvements

In StarOS releases prior to 14.0, there were only 10 PCC rules that were recovered per bearer in the event of a session manager crash. In 14.0 and later releases, this limit has been increased to 24. That is, up to 24 PCC rules can be recovered post ICSR.

With the increase in the limit of PCC rules that can be recovered, the rules are not lost and hence the charging applied to the end users are not impacted.

Impact on xDR File Naming

The xDR file name is limited to 256 characters with the following syntax:

basename_ChargSvcName_timestamp_SeqNumResetIndicator_FileSeqNumber

where:

- **basename**—A global configurable text string that is unique per system that uniquely identifies the global location of the system running ECS.
- **ChargSvcName**—A system context-based configurable text string that uniquely identifies a specific context-based charging service.
- **timestamp**—Date and time at the instance of file creation. Date and time in the form of “MMDDYYYYHHmmSS” where HH is a 24-hour value from 00-23.
- **SeqNumResetIndicator**—A one-byte counter used to discern the potential for duplicated FileSeqNumber with a range of 0 through 255, which is incremented by a value of 1 for the following conditions:
  - Failure of an ECS software process on an individual packet processing card
  - Failure of a system such that a second system takes over according to the Inter-chassis Session Recovery feature
  - File Sequence Number (FileSeqNumber) rollover from 999999999 to 0
**Enhanced Charging Service Overview**

- **FileSeqNumber**—Unique file sequence number for the file with nine-digit integer having range from 00000000 to 999999999. It is unique on each system.

With inter-chassis session recovery, only the first two fields in the xDR file names remain consistent between the active and standby chassis as these are parameters that are configured locally on the chassis. Per inter-chassis session recovery implementation requirements, the two chassis systems must be configured identically for all parameters not associated with physical connectivity to the distribution node.

The fields “timestamp”, “SeqNumResetIndicator”, and “FileSeqNumber” are all locally generated by the specific system through CDR subsystem, regardless of whether they are in an Inter-chassis Session Recovery arrangement or not.

- The “timestamp” value is unique to the system generating the actual xDRs and generated at the time the file is opened on the system.
- The SeqNumResetIndicator is a unique counter to determine the number of resets applied to FileSeqNumber. This counter is generated by CDR subsystem and increment the counter in event of resets in FileSeqNumber. This is required as “timestamp” field is not sufficient to distinguish between a unique and a duplicate xDR.

As such, the “SeqNumResetIndicator” field is used to distinguish between xDR files which have the same “FileSeqNumber” as a previously generated xDR as a result of:

- Normal operation, for example a rollover of the “FileSeqNumber” from maximum limit to 0.
- Due to a failure of one of the ECS processes running on a packet processing card card.
- Failure of the system (that is, Inter-chassis Session Recovery switchover).

In any scenario where the “FileSeqNumber” is reset to 0, the value of the “SeqNumResetIndicator” field is incremented by 1.

- The value of the “FileSeqNumber” is directly linked to the ECS process that is generating the specific xDRs. Any failure of this specific ECS process results in resetting of this field to 0.

**Impact on xDR File Content**

The following scenarios impact the xDR file content:

- On failure of an active chassis:

  On system startup, xDR files are generated in accordance with the standard processes and formats. If the system fails at any time it results in an inter-chassis session recovery switchover from active to standby and the following occurs depending on the state of the call/flow records and xDR file at the time of failure:

  - Call/flow records that were being generated and collected in system memory prior to being written out to /records directory on packet processing card RAM are not recoverable and therefore are lost.
  - Closed xDRs that have been written out to records directory on packet processing card RAM but that have yet to be retrieved by the ESS are recoverable.
  - Closed xDRs that have been retrieved and processed by the ESS have no impact.

- On the activation of a Standby chassis:

  Upon detection of a failure of the original active chassis, the standby chassis transits to the active state and begins serving the subscriber sessions that were being served by the now failed chassis. Any subsequent new subscriber session will be processed by this active chassis and will generate xDRs per the standard processes and procedures.

  However, this transition impacts the xDRs for those subscribers that are in-progress at the time of the transition. For in-progress subscribers, a subset of the xDR fields and their contents are carried over to the
newly active chassis via the SRP link. These fields and their contents, which are carried over after an Inter-chassis Session Recovery switchover, are as follows:

- HA-CORRELATION-ID
- PDSN-CORRELATION-ID (PDSN only)
- PDSN-NAS-IP-ADDRESS (PDSN only)
- PDSN-NAS-ID (PDSN only)
- USERNAME
- MSID
- RADIUS-NAS-IP-ADDRESS

All remaining fields are populated in accordance with the procedures associated with any new flow with the exceptions that, the field “First Packet Direction” is set to “Unknown” for all in-progress flows that were interrupted by the switchover and the field “FDR Reason” is marked as a PDSN Handoff and therefore is set to a value of “1” and corresponding actions are taken by the billing system to assure a proper and correct accounting of subscriber activities.
Chapter 2
Enhanced Charging Service Configuration

This chapter describes how to configure the Enhanced Charging Service (ECS) functionality, also known as Active Charging Service (ACS).

The following topics are covered in this chapter:

- Initial Configuration
- Configuring the Enhanced Charging Service
- Configuring Enhanced Features
Initial Configuration

Initial configuration includes the following:

**Step 1** Install the ECS license as described in the Installing the ECS License section.

**Step 2** Create the ECS administrative user account as described in the Creating the ECS Administrative User Account section.

**Step 3** Enable ECS as described in the Enabling Enhanced Charging Service section.

**Step 4** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command `save configuration`. For additional information on how to verify and save configuration files, refer to the System Administration Guide and the Command Line Interface Reference.

**Important:** Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the Command Line Interface Reference for complete information regarding all commands.

Creating the ECS Administrative User Account

At least one administrative user account with ECS privileges must be configured on the system. This is the account that is used to log on and execute ECS-related commands. For security purposes, it is recommended that these user accounts be created along with general system functionality administration.

To create the ECS administrative user account, use the following configuration:

```
configure

context local

   administrator <user_name> password <password> ecs

end
```

Notes:

- Aside from having ECS capabilities, an ECS Administrator account also has the same capabilities and privileges as any other system-level administrator account.

- You can also create system ECS user account for a config-administrator, operator, or inspector. ECS accounts have the same system-level privileges of normal system accounts except that they have full ECS command execution capability. For example, an ECS account has rights to execute every command that a regular administrator can in addition to all of the ECS commands.

- Note that only Administrator and Config-administrator level users can provision ECS functionality. Refer to the Configuring System Settings chapter of the System Administration and Configuration Guide for additional information on administrative user privileges.
Installing the ECS License

The ECS in-line service is a licensed Cisco product. Separate session and feature licenses may be required. Contact your Cisco account representative for detailed information on licensing requirements.

For information on installing and verifying licenses, refer to the Managing License Keys section of the Software Management Operations chapter in the System Administration Guide.

Enabling Enhanced Charging Service

Enhanced charging must be enabled before configuring charging services.

To enable Enhanced Charging Service, use the following configuration:

```
configure

require active-charging
context local

interface <interface_name>

  ip address <ip_address/mask>

  exit

server ftpd

end
```

Notes:

- The `require active-charging` command must be configured before any services are configured, so that the resource subsystem can appropriately reserve adequate memory for ECS-related tasks. After configuring this command, the configuration must be saved and the system rebooted in order to allocate the resources for ECS on system startup.
Configuring the Enhanced Charging Service

A charging service has analyzers that define which packets to examine and ruledefs (ruledefs) that define what packet contents to take action on and what action to take when the ruledef is matched. Charging services are configured at the global configuration level and are available to perform packet inspection on sessions in all contexts.

To configure the Enhanced Charging Service:

Step 1 Create the ECS service as described in the Creating the Enhanced Charging Service section.

Step 2 Configure a ruledef as described in the Configuring Rule Definitions section.

Step 3 Create a charging action as described in the Configuring Charging Actions section.

Step 4 Define a rulebase as described in the Configuring Rulebase section.

Step 5 Optional. Define a rulebase list in the ACS configuration mode and configure the rulebase list in an APN, as described in the Configuring Rulebase Lists section.

Step 6 Set EDR formats as described in the Setting EDR Formats section.

Step 7 Set UDR formats as described in the Setting UDR Formats section.

Step 8 Enable charging record retrieval as described in the Enabling Charging Record Retrieval section.

Step 9 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command `save configuration`. For additional information on how to verify and save configuration files, refer to the System Administration Guide and the Command Line Interface Reference.

**Important:** Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the Command Line Interface Reference for complete information regarding all commands.

Creating the Enhanced Charging Service

To create an Enhanced Charging Service, use the following configuration:

```
configure

    active-charging service <ecs_service_name>

end
```

Notes:

- In this release, only one ECS service can be created in a system.
Configuring Rule Definitions

This section describes how to create and configure rule definitions.

To create and configure a ruledef use the following configuration:

```
configure

active-charging service <ecs_service_name>

ruledef <ruledef_name>

<protocol> <expression> <operator> <condition>

rule-application { charging | post-processing | routing }

end
```

Notes:

- If the same ruledef is to be used for charging in one rulebase and for post-processing in another, then two separate identical ruledefs should be defined.
- For information on all the protocol types, expressions, operators, and conditions supported, refer to the ACS Ruledef Configuration Mode Commands chapter of the Command Line Interface Reference.
- The rule-application command specifies the ruledef type. By default, if not specified, the system considers a ruledef as a charging ruledef.
- In 14.1 and earlier releases, a maximum of 10 rule expressions (rule-lines) can be added in one ruledef. In 15.0 and later releases, a maximum of 32 rule expressions (rule-lines) can be added in one ruledef.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging ruledef { all | charging | name ruledef_name | post-processing | routing }
```

Configuring Group of Ruledefs

A group-of-ruledefs enables grouping rules into categories, so that charging systems can base the charging policy on the category.

To create and configure a group-of-ruledefs, use the following configuration:

```
configure

active-charging service <ecs_service_name>

group-of-ruledefs <ruledef_group_name>

add-ruledef priority <priority> ruledef <ruledef_name>
```
group-of-ruledefs-application { charging | content-filtering | gx-alias | post-processing }
end

Notes:
- A maximum of 128 ruledefs can be added to a group-of-ruledef.
- In 14.1 and earlier releases, a maximum of 64 group-of-ruledefs can be configured.
  In 15.0 and later releases, a maximum of 128 group-of-ruledefs can be configured.
- The group-of-ruledefs-application command specifies the group-of-ruledef type. By default, if not specified, the system considers a group-of-ruledef as a charging group-of-ruledef.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:
show active-charging group-of-ruledefs name <ruledef_group_name>

Configuring Charging Actions

Charging actions are used with rulebases and must be created before a rulebase is configured.
To create a charging action, use the following configuration:

configure

  active-charging service <ecs_service_name>

  charging-action <charging_action_name>

  content-id <content_id>

  retransmissions-counted

  billing-action { create-edrs { charging-edr <charging_edr_format_name> | reporting-edr <reporting_edr_format_name> } + [ wait-until-flow-ends ] | egcdr | exclude-from-udrs | radius | rf } +

end

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:
show active-charging charging-action name <charging_action_name>

Configuring IP Readdressing

Readdressing of packets based on the destination IP address of the packets enables redirecting unknown gateway traffic to known/trusted gateways. This is implemented by configuring the re-address server in the charging action.
To configure the IP Readdressing feature, use the following configuration:

```conf
configure
cactive-charging service <ecs_service_name>

charging-action <charging_action_name>

    flow action readdress server ipv4_address/ipv6_address [ discard-on-failure ] [ dns-proxy-bypass ] [ port port_number [ discard-on-failure ] [ dns-proxy-bypass ] ]

end
```

To configure the IP Readdressing feature when the readdress server-list is defined under charging-action, use the following configuration:

```conf
configure
active-charging service <ecs_service_name>

charging-action <charging_action_name>

    flow action readdress server-list server_list_name [ hierarchy ][ round-robin ] [ dns-proxy-bypass ] [ discard-on-failure ]

end
```

### Configuring Next Hop Address

To configure the Next Hop Address configuration feature, use the following configuration:

```conf
configure
active-charging service <ecs_service_name>

charging-action <charging_action_name>

    nexthop-forwarding-address <ip_address>

end
```

### Configuring Rulebase

A rulebase specifies which protocol analyzers to run and which packets are analyzed. Multiple rulebases may be defined for the Enhanced Charging Service. A rulebase is basically a subscriber’s profile in a charging service.

To create and configure a rulebase, use the following configuration:

```conf
configure
active-charging service <ecs_service_name>

    rulebase <rulebase_name>
```
flow end-condition { content-filtering | hagr | handoff | normal-end-signaling | session-end | url-blacklisting | timeout } [ flow-overflow ] + { charging-edr <charging_edr_format_name> | reporting-edr <reporting_edr_format_name> }

billing-records udr udr-format <udr_format_name>

action priority <action_priority> { [ dynamic-only | static-and-dynamic | timedef <timedef_name> ] [ group-of-ruledefs <ruledef_group_name> | ruledef <ruledef_name> ] charging-action <charging_action_name> [ monitoring-key <monitoring_key> ] [ description <description> ] }

route priority <route_priority> ruledef <ruledef_name> analyzer <analyzer> [ description <description> ]

rtp dynamic-flow-detection

udr threshold interval <interval>

cca radius charging context <context> group <group_name>

cca radius accounting interval <interval>

end

Notes:
- When R7 Gx is enabled, “static-and-dynamic” rules behave exactly like “dynamic-only” rules. I.e. they must be activated explicitly by the PCRF. When Gx is not enabled, “static-and-dynamic” rules behave exactly like static rules.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

show active-charging rulebase name <rulebase_name>

Configuring Rulebase Lists

To create a rulebase list, use the following configuration:

configure

active-charging service <ecs_service_name>

rulebase-list <rulebase_list_name> <rulebase_name> [ <rulebase_name> + ]

exit

Configuring a Rulebase List in an APN

To configure the rulebase list that was created in the ACS configuration mode in an APN, use the following configuration:
configuring the enhanced charging service

<table>
<thead>
<tr>
<th>Enhanced Charging Service Configuration</th>
<th>Configuring the Enhanced Charging Service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enhanced Charging Service Configuration</strong></td>
<td><strong>Configuring the Enhanced Charging Service</strong></td>
</tr>
<tr>
<td><strong>Cisco ASR 5x00 Enhanced Charging Services Administration Guide</strong></td>
<td><strong>Cisco ASR 5x00 Enhanced Charging Services Administration Guide</strong></td>
</tr>
</tbody>
</table>

**Verifying your configuration**

To verify your configuration for the rulebase list and APN, in the Exec mode, enter the following command:

```
show configuration
```

To verify your APN configuration, in the Exec mode, enter the following command:

```
show configuration apn <apn_name>
```

**Setting EDR Formats**

ECS generates postpaid charging data files which can be retrieved from the system periodically and used as input to a billing mediation system for postprocessing.

EDRs are generated according to action statements in rule commands.

Up to 32 different EDR schema types may be specified, each composed of up to 32 fields or analyzer parameter names. The records are written at the time of each rule event in a comma-separated (CSV) format.

**Important:** If you have configured RADIUS Prepaid Billing, configuring charging records is optional.

To set the EDR formats use the following configuration:

```
configure

context <context_name>
apn <apn_name>
active-charging rulebase-list <rulebase_list_name>
exit
```

**Important:** For information on EDR format configuration and rule variables, refer to the EDR Format Configuration Mode Commands chapter of the Command Line Interface Reference.

```
configure

active-charging service <ecs_service_name>
edr-format <edr_format_name>

    attribute <attribute> { [ format { MM/DD/YY-HH:MM:SS | MM/DD/YYYY-HH:MM:SS | YYYY/MM/DD-HH:MM:SS | YYYYMMDDHHMMSS | seconds } [ localtime ] | [ { ip | tcp } { bytes | pkts } { downlink | uplink } ] priority <priority> }

    rule-variable <protocol> <rule> priority <priority>

    event-label <event_label> priority <priority>

end
```

**Important:** For information on EDR format configuration and rule variables, refer to the EDR Format Configuration Mode Commands chapter of the Command Line Interface Reference.
Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging edr-format name <edr_format_name>
```

Setting UDR Formats

ECS generates postpaid charging data files which can be retrieved from the system periodically and used as input to a billing mediation system for postprocessing.

UDRs are generated according to action statements in rule commands. Up to 32 different UDR schema types may be specified, each composed of up to 32 fields or analyzer parameter names. The records are written thresholds in a comma-separated (CSV) format.

**Important:** If you have configured RADIUS Prepaid Billing, configuring charging records is optional.

To set the UDR format, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
  udr-format <udr_format_name>
    attribute <attribute> { [ format { MM/DD/YY-HH:MM:SS | MM/DD/YYYY-HH:MM:SS | YYYY/MM/DD-HH:MM:SS | YYYYMMDDHHMSS | seconds } ] [ localtime ] | [ { bytes | pkts } { downlink | uplink } ] } priority <priority>
  end
```

**Important:** For information on UDR format configuration and rule variables, refer to the UDR Format Configuration Mode Commands chapter of the Command Line Interface Reference.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging udr-format name <udr_format_name>
```

Enabling Charging Record Retrieval

To retrieve charging records you must configure the context that stores the charging records to accept SFTP connections.

To enable SFTP, use the following configuration:

```
configure
  context local
```
administrator <user_name> [ encrypted ] password <password>

config-administrator <user_name> [ encrypted ] password <password>

exit

crash context <context_name>

ssh generate key

server sshd

subsystem sftp

end

Notes:
- You must specify the sftp keyword to enable the new account to SFTP into the context to retrieve record files.

Optional Configurations

This section describes the following optional configuration procedures:

- Configuring a Rulebase for a Subscriber
- Configuring a Rulebase within an APN
- Configuring Charging Rule Optimization

Important: Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the Command Line Interface Reference for complete information regarding all commands.

Configuring a Rulebase for a Subscriber

This section describes how to apply an existing rulebase to a subscriber.

For information on how to configure rulebases, see the Configuring Rulebases section.

To configure a rulebase for a subscriber, use the following configuration:

configure

crash context <context_name>

subscriber name <subscriber_name>

active-charging rulebase <rulebase_name>

end
Configuring a Rulebase within an APN

This section describes how to configure an existing rulebase within an APN for a GGSN. For information on how to configure rulebases, see the Configuring Rulebases section.

Important: This information is only applicable to GGSN networks.

To configure a rulebase in an APN, use the following configuration:

```plaintext
configure
    context <context_name>
    apn <apn_name>
    active-charging rulebase <rulebase_name>
end
```

Configuring Charging Rule Optimization

This section describes how to configure the internal optimization level for improved performance when the system evaluates each instance of the action CLI command.

To configure the rule optimization level, use the following configuration:

```plaintext
configure
    active-charging service <ecs_service_name>
    rulebase <rulebase_name>
    charging-rule-optimization { high | low | medium }
end
```

Notes:

- In StarOS 14.0 and later releases, the charging-rule-optimization command is deprecated. Rule optimization is always enabled with the optimization level set to high as standard behavior.
- In 11.0 and later releases, the medium option is deprecated.
- Both the high and medium options cause reorganization of the entire memory structure whenever any change is made (for example, addition of another action CLI command).
- The high option causes allocation of a significant amount of memory for the most efficient organization.
Configuring Enhanced Features

The configuration examples in this section are optional and provided to cover the most common uses of ECS in a live network.

The following topics are covered in this section:

- Configuring Prepaid Credit Control Application (CCA)
- Configuring Redirection of Subscriber Traffic to ECS
- Configuring GTPP Accounting
- Configuring DNS Snooping Feature
- Configuring EDRUDR Parameters
- Configuring Post Processing Feature
- Configuring Service Group QoS Feature
- Configuring TCP Proxy
- Configuring Tethering Detection Feature
- Configuring Time-of-Day Activation/Deactivation of Rules Feature
- Configuring Transactional Rule Matching Feature
- Configuring Flow Aware Packet Acceleration Feature
- Configuring Retransmissions Under Rulebase or Service Level CLI
- Configuring Websockets
- Configuring URL Filtering Feature
- Configuring x-header Insertion and Encryption Feature

Configuring Prepaid Credit Control Application (CCA)

This section describes how to configure the Prepaid Credit Control Application for Diameter or RADIUS.

**Important:** To configure and enable Diameter and DCCA functionality with ECS, you must obtain and install the relevant license on the chassis. Contact your Cisco account representative for detailed information on licensing requirements.

**Important:** Before configuring Diameter or RADIUS CCA, you must configure AAA parameters. For more information, if you are using StarOS 12.3 or an earlier release, refer to the *AAA and GTPP Interface Administration and Reference*. If you are using StarOS 14.0 or a later release, refer to the *AAA Interface Administration and Reference*.

To configure Prepaid Credit Control Application:

**Step 1** Configure the Prepaid Credit Control Application for Diameter or RADIUS as described in the *Configuring Prepaid CCA for Diameter or RADIUS* section.
Step 2 Configure the required Prepaid Credit Control Mode:

- Configuring Diameter Prepaid Credit Control Application (DCCA)
- Configuring RADIUS Prepaid Credit Control Application

**Important:** Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the Command Line Interface Reference for complete information regarding all commands.

### Configuring Prepaid CCA for Diameter or RADIUS

To configure the Prepaid Credit Control Application for Diameter or RADIUS, use the following configuration:

```plaintext
configure
  active-charging-service <ecs_service_name>
  charging-action <charging_action_name>
    cca charging credit [ preemptively-request | rating-group <coupon_id> ]
  exit
  credit-control [ group <group_name> ]
    mode { diameter | radius }
    quota time-threshold { <absolute_value> | percent <percent_value> }
    quota unit-threshold { <absolute_value> | percent <percent_value> }
    quota volume-threshold { <absolute_value> | percent <percent_value> }
  end
```

Notes:

- `<ecs_service_name>` must be the name of the Enhanced Charging Service in which you want to configure Prepaid Credit Control Application.
- `<charging_action_name>` must be the name of the charging action for which you want to configure Prepaid Credit Control Application.
- **Optional:** To configure the redirection of URL for packets that match a ruledef and action on quota request timer, in the Charging Action Configuration Mode, enter the following command. This command also specifies the redirect-URL action on packet and flow for Session Control functionality.

In 12.2 and later releases:

```plaintext
flow action redirect-url <redirect_url> [ [ encryption { blowfish128 | blowfish64 } | { aes128 | aes256 } [salt] ] [ encrypted ]
  key <key> ] [ clear-quota-retry-timer ] [ first-request-only [ post-redirect { allow | discard | terminate } ] ] ]
```

In 12.1 and earlier releases:

```plaintext
flow action redirect-url <redirect_url> [ clear-quota-retry-timer ]
```
The following example shows the redirection of a URL for packets that match a ruledef:

```
charging-action http-redirect
content-id 3020
retransmissions-counted
billing-action exclude-from-udrs
flow action redirect-url "http://10.1.67.214/cgi-bin/aoc.cgi\077
imsi=#bearer.calling-station-id#&url=#http.url#&acctsessid=#bearer.acct-
session-id#&correlationid=#bearer.correlation-id#&username=#bearer.user-
name#&ip=#bearer.served-bsa-addr#&subid=#bearer.subscribe-

end
```

- **Optional:** To configure credit control quota related parameters, use the following configuration:

```
configure
active-charging service <ecs_service_name>
rulebase <rulebase_name>
cca quota { holding-time <holding_time> content-id <content_id> | retry-time <retry_time> [ max-retries <max_retries> ] }
cca quota time-duration algorithm { consumed-time <consumed_time> [ plus-idle ] [ content-id <content_id> ] | continuous-time-periods <seconds> [ content-id <content_id> ] | parking-meter <seconds> [ content-

end
```

<rulebase_name> must be the name of the rulebase in which you want to configure Prepaid Credit Control configurables.

- **Optional:** To define credit control rules for quota state and URL redirect match rules with RADIUS AVP, use the following configuration:

```
configure
active-charging service <ecs_service_name>
ruledef <ruledef_name>
cca quota-state <operator> { limit-reached | lower-bandwidth }
cca redirect-indicator <operator> <indicator_value>
end
```

<ruledef_name> must be the name of the ruledef that you want to use for Prepaid Credit Control Application rules.
**cca redirect-indicator** configuration is a RADIUS-specific configuration.

- **Optional:** This is a Diameter-specific configuration. To configure the failure handling options for credit control session, in the Credit Control Configuration Mode, use the following configuration:

```
configure

active-charging service <ecs_service_name>

credit-control [ group <group_name> ]


end
```

- **Optional:** To configure the triggering option for credit reauthorization when the named values in the subscriber session changes, use the following configuration:

```
configure

active-charging service <ecs_service_name>

credit-control [ group <group_name> ]

  trigger type { cellid | lac | qos | rat | sgsn } +

end
```

- **Optional:** This is a Diameter-specific configuration. If the configuration is for 3GPP network, to configure the virtual or real APN name to be sent in Credit Control Application (CCA) message, use the following configuration:

```
configure

active-charging service <ecs_service_name>

credit-control [ group <group_name> ]

  apn-name-to-be-included { gn | virtual }

end
```

**Configuring Diameter Prepaid Credit Control Application (DCCA)**

This section describes how to configure the Diameter Prepaid Credit Control Application.

---

**Important:** To configure and enable Diameter and DCCA functionality with ECS, you must obtain and install the relevant license on the chassis. Contact your Cisco account representative for detailed information on licensing requirements.
Enhanced Charging Service Configuration

Configuring Enhanced Features

**Important:** It is assumed that you have already fully configured the AAA parameters, and Credit Control Application as described in Configuring Prepaid Credit Control Application (CCA) section for Diameter mode. For information on configuring AAA parameters, if you are using StarOS 12.3 or an earlier release, refer to the AAA and GTPP Interface Administration and Reference. If you are using StarOS 14.0 or a later release, refer to the AAA Interface Administration and Reference.

To configure Diameter Prepaid Credit Control Application, use the following configuration.

```plaintext
configure

active-charging service <ecs_service_name>

credit-control [ group <cc_group_name> ]

mode diameter

diameter origin endpoint <endpoint_name>

diameter dictionary <dcca_dictionary>

diameter peer-select peer <peer_name> [ realm <realm_name> ] [ secondary-peer <sec_peer_name> [ realm <realm_name> ] ] [  
imi-based { [ prefix | suffix ] <imi/prefix/suffix_start_value> } [ to <imi/prefix/suffix_end_value> ] ]

end
```

Notes:

- Diameter peer configuration set with the `diameter peer-select` command can be overridden by the `dcca peer-select peer` command in the APN Configuration mode for 3GPP service networks, and in Subscriber Configuration mode in other service networks.

- The specific Credit Control Group to be used for subscribers must be configured in the APN Configuration Mode using the `credit-control-group <cc_group_name>` command.

- **Optional:** To configure the maximum time, in seconds, to wait for a response from Diameter peer, in the Credit Control Configuration Mode, enter the following command:

```
diameter pending-timeout <duration>
```

- **Optional:** To configure Diameter Credit Control Session Failover, in the Credit Control Configuration Mode, enter the following command:

```
diameter session failover
```

When enabled, in the event of failure, failure handling action is based on the `failure-handling` CLI.

- **Optional:** If you want to configure the service for IMS authorization in 3GPP service network, you can configure dynamic rule matching with Gx interface and dynamic rule matching order in rulebase, use the following configuration:

```
Configure

active-charging service <ecs_service_name>

rulebase <rulebase_name>

dynamic-rule order { always-first | first-if-tied }
```
action priority <action_priority> { [ dynamic-only | static-and-
  dynamic | timedef <timedef_name> ] } [ group-of-ruledefs
<ruledef_group_name> | ruledef <ruledef_name> ] } charging-action
<charging_action_name> [ monitoring-key <monitoring_key> ] [ description
<description> ] }

counters <counter_name> 

• Optional: To configure Diameter group AVP Requested-Service-Unit for Gy interface support to include a sub-
  AVP in CCRs using volume, time, and unit specific charging, in the Rulebase Configuration Mode, enter the
  following command:

  cca diameter requested-service-unit sub-avp { 
  time cc-time <duration> | units cc-
  service-specific-units <charging_unit> | volume { cc-input-octets <bytes> | cc-
  output-octets <bytes> | cc-total-octets <bytes> } + }

• Ensure the Diameter endpoint parameters are configured. For information on configuring Diameter endpoint, if
  you are using StarOS 12.3 or an earlier release, refer to the AAA and GTPP Interface Administration and
  Reference. If you are using StarOS 14.0 or a later release, refer to the AAA Interface Administration and
  Reference.

Configuring Peer-Select in Subscriber Configuration Mode (Optional)

This section describes how to configure Diameter peer-select within a subscriber configuration.

> Important: The dcca peer-select configuration completely overrides all instances of diameter peer-
select configured within the Credit Control Configuration Mode for an Enhanced Charging Service.

To configure DCCA peers within a subscriber configuration, use the following configuration:

configure

  context <context_name>

  subscriber name <subscriber_name>

  dcca peer-select peer <host_name> [ [ realm <realm_name> ] [ secondary-peer
<host_name> [ realm <realm_name> ] ] ]}

end

Configuring Peer-Select in APN Configuration Mode (Optional)

This section describes how to configure Diameter peer-select within an APN configuration.

> Important: This information is only applicable to GGSN networks.

> Important: The dcca peer-select configuration completely overrides all instances of diameter peer-
select configured within the Credit Control Configuration Mode for an Enhanced Charging Service.

To configure DCCA peers within an APN, use the following configuration:
Configuring Enhanced Features

Cisco ASR 5x00 Enhanced Charging Services Administration Guide

Enhanced Charging Service Configuration

Configuring RADIUS Prepaid Credit Control Application

RADIUS prepaid billing operates on a per content-type basis. Individual content-types are marked for prepaid treatment. When a traffic analysis rule marked with prepaid content-types matches, it triggers prepaid charge management.

**Important:** The RADIUS Prepaid feature of ECS has no connection to the system-level Prepaid Billing Support or the 3GPP2 Prepaid features that are enabled under different licenses.

**Important:** It is assumed that you have already fully configured the AAA parameters, and Credit Control Application as described in Configuring Prepaid Credit Control Application (CCA) section for RADIUS mode. For information on configuring AAA parameters, if you are using StarOS 12.3 or an earlier release, refer to the AAA and GTPP Interface Administration and Reference. If you are using StarOS 14.0 or a later release, refer to the AAA Interface Administration and Reference.

To configure RADIUS Prepaid Charging with Enhanced Charging, use the following configuration.

```bash
configure

context <context_name>

apn <apn_name>

    dcca peer-select peer <host_name> [ [ realm <realm_name> ] [ secondary-peer <host_name> [ realm <realm_name> ] ] ]

end
```

**Notes:**

- `<rulebase_name>` must be the name of the rulebase in which you want to configure Prepaid Credit Control configurables.
- `<vpn_context>` must be the charging context in which the RADIUS parameters are configured.
- **Optional:** To specify the accounting interval duration for RADIUS prepaid accounting, in the Rulebase Configuration Mode, enter the following command:

```bash
cca radius accounting interval <interval>
```
• Optional: To specify the user password for RADIUS prepaid services, in the Rulebase Configuration Mode, enter the following command:
  
  \texttt{cca radius user-password [ encrypted ] password <password>}

• Ensure the RADIUS server parameters are configured. For more information, if you are using StarOS 12.3 or an earlier release, refer to the \textit{AAA and GTPP Interface Administration and Reference}. If you are using StarOS 14.0 or a later release, refer to the \textit{AAA Interface Administration and Reference}.

### Configuring Redirection of Subscriber Traffic to ECS

User traffic is directed through the ECS service inspection engine by using Access Control List (ACL) mechanism to selectively steer subscriber traffic.

To configure redirection of subscriber traffic to ECS:

**Step 1** Create an ECS ACL as described in the \textit{Creating an ECS ACL} section.

**Step 2** Apply an ACL to an individual subscriber as described in the \textit{Applying an ACL to an Individual Subscriber} section.

**Step 3** Apply an ACL to the subscriber named default as described in the \textit{Applying an ACL to the Subscriber Named default} section.

**Step 4** Apply the ACL to an APN as described in the \textit{Applying the ACL to an APN} section.

**Important:** Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the \textit{Command Line Interface Reference} for complete information regarding all commands.

### Creating an ECS ACL

To create an ACL to use in steering subscriber traffic through ECS, use the following configuration:

```plaintext
configure
  context <context_name>
  ip access-list <access_list_name>
    redirect css service <ecs_service_name> <keywords> <options>
  end
```

**Notes:**

• \texttt{<ecs_service_name>} must be the enhanced charging service’s name; no CSS service needs to be configured.

### Applying an ACL to an Individual Subscriber

IP ACLs are applied to subscribers via attributes in their profile. The subscriber profile could be configured locally on the system or remotely on a RADIUS server.
To apply an ACL to a RADIUS-based subscriber, use the Filter-Id attribute. For more information on this attribute, if you are using StarOS 12.3 or an earlier release, refer to the *AAA and GTPP Interface Administration and Reference*. If you are using StarOS 14.0 or a later release, refer to the *AAA Interface Administration and Reference*.

To apply an ACL to an individual subscriber, use the following configuration:

```
configure
  context <context_name>
    subscriber name <subscriber_name>
      ip access-group <acl_name> [ in | out ]
    end
```

**Applying an ACL to the Subscriber Named default**

To apply an ACL to the default subscriber, use the following configuration:

```
configure
  context <context_name>
    subscriber default
      ip access-group <acl_name> [ in | out ]
    end
```

**Applying the ACL to an APN**

To apply an ACL to an APN, use the following configuration:

```
configure
  context <context_name>
    apn <apn_name>
      ip access-group <acl_name> [ in | out ]
    end
```

**Important:** This information is only applicable to UMTS networks.

```
configure
  context <context_name>
    apn <apn_name>
      ip access-group <acl_name> [ in | out ]
    end
```

**Configuring GTPP Accounting**

For information on configuring GTPP accounting, if you are using StarOS 12.3 or an earlier release, refer to the *AAA and GTPP Interface Administration and Reference*. If you are using StarOS 14.0 or a later release, refer to the *AAA Interface Administration and Reference*. 
Configuring DNS Snooping Feature

This section describes how to configure the DNS Snooping feature.

To configure the DNS Snooping feature use the following configuration:

```
configure
  active-charging service <ecs_service_name>
  ip dns-learnt-entries timeout <timeout_period>
  ruledef <ruledef_name>
    ip server-domain-name { = | contains | ends-with | starts-with }
<domain_name/domain_name_segment>
    ...  
  exit
  rulebase <rulebase_name>
    action priority <priority> ruledef <ruledef_name> charging-action
<charging_action_name>
    ...  
end
```

Configuring EDR/UDR Parameters

This section provides an example configuration to configure EDR/UDR file transfer and file properties parameters, including configuring hard disk support on SMC card on ASR 5000, transfer modes, transfer interval, etc.

To configure EDR/UDR file parameters:

```
configure
  context <context_name>
    edr-module active-charging-service [ charging | reporting ]
    cdr { purge { storage-limit <storage_limit> | time-limit <time_limit> } [ max-files <max_records_to_purge> ] | push-interval <push_interval> | push-trigger space-usage-percent <trigger_percentage> | remove-file-after-transfer | transfer-mode { pull [ module-only ] | push primary { encrypted-url <encrypted_url> | url <url> } [ [ max-files <max_records> ] [ module-only ] [ secondary { encrypted-secondary-url <encrypted_secondary_url> | secondary-url <secondary_url> } ] [ via local-context ] + ] | use-harddisk }
```
file [ charging-service-name { include | omit } ] [ compression { gzip | none } ] [ current-prefix <string> ] [ delete-timeout <seconds> ] [ directory <directory_name> ] [ edr-format-name ] [ exclude-checksum-record ] [ field-separator { hyphen | omit | underscore } ] [ file-sequence-number rulebase-seq-num ] [ headers ] [ name <file_name> ] [ reset-indicator ] [ rotation [ num-records <number> | time <seconds> | volume <bytes> ] ] [ sequence-number { length <length> | omit | padded | padded-six-length | unpadded } ] [ storage-limit <limit> ] [ single-edr-format ] [ time-stamp { expanded-format | rotated-format | unix-format } ] [ trailing-text <string> ] [ trap-on-file-delete ] [ xor-final-record ] +

exit

udr-module active-charging-service

file [ charging-service-name { include | omit } ] [ compression { gzip | none } ] [ current-prefix <string> ] [ delete-timeout <seconds> ] [ directory <directory_name> ] [ exclude-checksum-record ] [ field-separator { hyphen | omit | underscore } ] [ file-sequence-number rulebase-seq-num ] [ headers ] [ name <file_name> ] [ reset-indicator ] [ rotation [ num-records <number> | time <seconds> | volume <bytes> ] ] [ sequence-number { length <length> | omit | padded | padded-six-length | unpadded } ] [ storage-limit <limit> ] [ time-stamp { expanded-format | rotated-format | unix-format } ] [ trailing-text <string> ] [ trap-on-file-delete ] [ udr-seq-num ] [ xor-final-record ] +

end

Notes:
- The `cdr` command keywords can be configured either in the EDR or the UDR Configuration Mode. Configuring in one mode prevents the configurations from being applied in the other mode.
- If the `edr-module active-charging-service` command is configured without the `charging` or `reporting` keywords, by default the EDR module is enabled for charging EDRs.
- When the configured threshold limit is reached on the hard disk drive, the records that are created dynamically in the `/mnt/hd-raid/data/records/` directory are automatically deleted. Files that are manually created should be deleted manually.
- The `use-harddisk` keyword is only available on the ASR 5000.

Verifying your Configurations

To view EDR-UDR file statistics, in the Exec Mode, enter the following command:

```
show active-charging edr-udr-file statistics
```

Pushing EDR/UDR Files Manually

To manually push EDR/UDR files to the configured L-ESS, in the Exec mode, use the following command:

```
cdr-push { all | local-filename <file_name> }
```

Notes:
- Before you can use this command, the CDR transfer mode and file locations must be set to push in the EDR/UDR Module Configuration Mode.
- The `cdr-push` command is available in the Exec Mode.
• `<file_name>` must be absolute path of the local file to push.

Retrieving EDR and UDR Files

To retrieve UDR or EDR files you must SFTP into the context that was configured for EDR or UDR file generation. This was done with the FTP-enabled account that you configured in the Enabling Charging Record Retrieval section. The following commands use SFTP to log on to a context named ECP as a user named `ecpadmin`, through an interface configured in the ECS context that has the IP address 192.168.1.10 and retrieve all EDR or UDR files from the default locations:

```
sftp -oUser=ecpadmin@ECP 192.168.1.10:/records/edr/*
sftp -oUser=ecpadmin@ECP 192.168.1.10:/records/udr/*
```

Configuring Post Processing Feature

This section describes how to configure the Post-processing feature to enable processing of packets even if rule matching for them has been disabled.

To configure the Post-processing feature, use the following configuration:

```
configure

    active-charging service <ecs_service_name>

    ruledef <ruledef_name>

        <protocol> <expression> <operator> <condition>

        rule-application post-processing

        exit

        charging-action <charging_action_name>

        ...

        exit

    rulebase <rulebase_name>

        action priority <action_priority> { [ dynamic-only | static-and-dynamic | timedef <timedef_name> ] [ group-of-ruledefs <ruledef_group_name> | ruledef <ruledef_name> ] charging-action <charging_action_name> [ monitoring-key <monitoring_key> ] [ description <description> ] }

    post-processing priority <priority> ruledef <ruledef_name> charging-action <charging_action_name>

    ...

end
```
Notes:

- In the Rulebase Configuration Mode, the ruledef configured for post-processing action must have been configured for post processing in the Ruledef Configuration Mode.
- If the same ruledef is required to be a charging rule in one rulebase and a post-processing rule in another rulebase, then two separate identical ruledefs must be defined.
- In this release, post processing with group-of-ruledefs is not supported.
- In this release, delay charging with dynamic rules is not supported, hence there cannot be dynamic post-processing rules.

**Configuring Service Group QoS Feature**

To create and configure a QoS-Group-of-Ruledefs, use the following configuration:

```plaintext
configure

active-charging service <ecs_service_name>

qos-group-of-ruledefs <qos_group_of_ruledefs_name> [ -noconfirm ] [ description <description> ]

add-ruledef <ruledef_name>

end
```

Notes:

- To configure flow action in the charging-action, in the ACS Charging Action Configuration Mode, use the `flow action` CLI command.
- To configure bandwidth limits for a flow, in the ACS Charging Action Configuration Mode use the `flow limit-for-bandwidth` CLI command.
- To view subscriber statistics and information on dynamic updates to charging parameters per call ID, in the Exec Mode, use the following command:

```plaintext
show active-charging subscribers callid <call_id> charging-updates [ statistics ] [ charging-action [ name <charging_action_name> ] | qos-group [ name <qos_group_of_ruledefs_name> ] ] [ | { grep <grep_options> | more } ]
```

**Configuring TCP Proxy**

This section describes how to configure the TCP Proxy feature.

To enable and configure the TCP Proxy feature in the rulebase, use the following configuration:

```plaintext
configure

active-charging service <ecs_service_name>

rulebase <rulebase_name>
```
Configuring Enhanced Features

```plaintext
tcp proxy-mode { dynamic { all | content-filtering | dcca | ip-readdressing |
next-hop-readdressing | xheader-insert + } | static { port [ <port_number> [ to
<port_number> ] ] } }
```

Configuring Flow Admission Control

To configure the TCP Proxy Flow Admission Control feature, use the following configuration:

```plaintext
configure
active-charging service <ecs_service_name>
fair-usage tcp-proxy max-flows-per-subscriber <max_flows>
fair-usage tcp-proxy memory-share <memory_share>
end
```

Notes:
- It is not necessary for the Fair Usage feature to be enabled before this configuration.
- `<max_flows>` specifies the maximum number of flows for which TCP Proxy can be used per subscriber. Note that this limit is per Session Manager.
- `<memory_share>` specifies what portion of ECS memory should be reserved for TCP Proxy flows. Note that it is a percentage value.

Verifying your Configuration

To verify your configuration, in the Exec mode, use the following command:

```plaintext
show active-charging tcp-proxy statistics [ rulebase <rulebase_name> ] [ verbose ] [ | { grep <grep_options> | more } ]
```

Configuring Tethering Detection Feature

This section describes how to configure the Tethering Detection feature to detect subscriber flows from PC devices tethered to mobile smartphones.

**Important:** This command is available only if the Smartphone Tethering Detection license is enabled. For more information, please contact your Cisco account representative.

To enable and configure the Tethering Detection feature, use the following configuration:

```plaintext
configure
active-charging service <ecs_service_name>
tethering-database [ os-signature <os_signature_db_file_name> | tac
<tac_db_file_name> | ua-signature <ua_signature_db_file_name> ] +
```
Upgrading Tethering Detection Databases

To upgrade the Tethering Detection feature databases, in the Exec mode, use the following CLI command:

```
upgrade tethering-detection database { all | os-signature | tac | ua-signature } [-noconfirm ]
```

Notes:
- To load and upgrade the databases used in detecting tethering, the database files must be copied from MUR onto the ASR chassis to the designated directory path for storing the database files:
  
  `/mnt/hd-raid/data/databases/`
  
  Any further upgrades to the database files can be done by placing the file named `new-filename` in the designated directory path. ECS auto-detects the presence of files available for upgrade daily. When a new version of a file is found, the upgrade process is triggered. The upgrade can also be forced by running the upgrade command in the CLI. On a successful upgrade this file is renamed to `filename`.

Sample Tethering Detection Feature Configurations

The following examples illustrate two different implementations of the Tethering Detection feature’s configuration.

- The following type of configuration is suitable where ECS performance is critical and the operator wants to put in a flat charging plan in place for all the tethered traffic. In such a scenario, addition of a single new ruledef to the configuration suffices. Placing this ruledef at the highest priority in the rulebase will ensure all the tethered flows are charged as per the tariff plan for tethered traffic.

  ```
  configure

  active-charging service ecs_service

  tethering-database

  ruledef tethered-traffic

  tethering-detection flow=tethered

  tcp any-match = TRUE
  ```
exit
ruledef ftp-pkts
  ftp any-match = TRUE
exit
ruledef http-pkts
  http any-match = TRUE
exit
ruledef tcp-pkts
  tcp any-match = TRUE
exit
ruledef ip-pkts
  ip any-match = TRUE
exit
ruledef http-port
  tcp either-port = 80
  rule-application routing
exit
ruledef ftp-port
  tcp either-port = 21
  rule-application routing
exit
charging-action premium
  content-id 1
  retransmissions-counted
  billing-action egcdr
exit
charging-action standard
  content-id 2
  retransmissions-counted
The following type of configuration is suitable when operators want to apply differentiated charging to various flows that are found to be tethered. In this case, traffic that requires different charging action or content ID when it is tethered will be identified using two ruledefs, one with “flow-is-tethered = TRUE” option and another without this option. This configuration provides finer granularity of control but results in higher performance degradation because the rule matching tree size increases.

```
configure

active-charging service ecs_service
tethering-database
ruledef ftp-pkts
   ftp any-match = TRUE
   exit
ruledef ftp-pkts-tethered
   ftp any-match = TRUE
tethering-detection flow-tethered
   exit
ruledef http-pkts
   http any-match = TRUE
```
exit
ruledef http-pkts-tethered
  http any-match = TRUE
  tethering-detection flow-tethered
  exit
ruledef tcp-pkts
  tcp any-match = TRUE
  exit
ruledef tcp-pkts-tethered
  tcp any-match = TRUE
  tethering-detection flow-tethered
  exit
ruledef ip-pkts
  ip any-match = TRUE
  exit
ruledef ip-pkts-tethered
  ip any-match = TRUE
  tethering-detection flow-tethered
  exit
ruledef http-port
  tcp either-port = 80
  rule-application routing
  exit
ruledef ftp-port
  tcp either-port = 21
  rule-application routing
  exit
charging-action premium-http
  content-id 10
retransmissions-counted
billing-action egcdr
exit
charging-action premium-ftp
content-id 20
retransmissions-counted
billing-action egcdr
exit
charging-action premium
content-id 1
retransmissions-counted
billing-action egcdr
exit
charging-action standard
content-id 2
retransmissions-counted
billing-action egcdr
exit
rulebase consumer
tethering-detection
action priority 10 ruledef ftp-pkts-tethered charging-action
premium-ftp
action priority 20 ruledef ftp-pkts charging-action standard
action priority 30 ruledef http-pkts-tethered charging-action
premium-http
action priority 40 ruledef http-pkts charging-action standard
action priority 50 ruledef tcp-pkts-tethered charging-action
premium
action priority 60 ruledef tcp-pkts charging-action standard
action priority 70
ruledef ip-pkts-tethered
charging-action premium

action priority 80
ruledef ip-pkts
charging-action standard

route priority 80
ruledef http-port analyzer http

exit

rulebase default
end

Configuring Time-of-Day Activation/Deactivation of Rules Feature

This section describes how to configure the Time-of-Day Activation/Deactivation of Rules feature to enable charging according to day/time.

To configure the Time-of-Day Activation/Deactivation of Rules feature, use the following configuration:

```
configure

active-charging service <ecs_service_name>
ruledef <ruledef_name>

... exit

timedef <timedef_name>

start day { friday | monday | saturday | sunday | thursday | tuesday | wednesday }
time <hh> <mm> <ss>
end day { friday | monday | saturday | sunday | thursday | tuesday | wednesday }
time <hh> <mm> <ss>

start time <hh> <mm> <ss>
end time <hh> <mm> <ss>

exit

charging-action <charging_action_name>

... exit

rulebase <rulebase_name>

action priority <action_priority>
timedef <timedef_name> [ group-of-ruledefs <ruledef_group_name> ]
ruledef <ruledef_name> ]
charging-action <charging_action_name> [ description <description> ]

... exit

end
```
Notes:
- In a timeslot if only the time is specified, that timeslot will be applicable for all days.
- If for a timeslot, “start time” > “end time”, that rule will span the midnight. I.e. that rule is considered to be active from the current day till the next day.
- If for a timeslot, “start day” > “end day”, that rule will span over the current week till the end day in the next week.
- In the following cases a rule will be active all the time:
  - A timedef is not configured in an action priority
  - A timedef is configured in an action priority, but the named timedef is not defined
  - A timedef is defined but with no timeslots

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging timedef name <timedef_name>
```

Configuring Transactional Rule Matching Feature

To enable the Transactional Rule Matching (TRM) feature, use the following configuration:

```
configure

active-charging service <ecs_service_name>

  rulebase <rulebase_name>

    transactional-rule-matching

end
```

Notes:
- Use the `no transactional-rule-matching` command or `default transactional-rule-matching` command to disable transactional rule matching.
- Transactional rule matching is disabled by default.

**Important:** From 16.0 release, Transactional Rule Matching and Fastpath functionalities have been merged, and will be governed by only the `transactional-rule-matching` keyword alone. The keyword `fastpath` independently can no longer be used to turn on or turn off this functionality.

To verify your configuration, in the Exec mode, enter the following command:

```
show active-charging rulebase name <rulebase_name>
```
Configuring Flow Aware Packet Acceleration Feature

To enable the Flow Aware Packet Acceleration feature, use the following configuration:

```plaintext
configure
  active-charging service <service_name>
    [no] accelerate-flow
  end
```

To verify your configuration, in the Exec mode, enter the following command:

`show active-charging accelerate-flow`

Configuring Retransmissions Under Rulebase or Service Level CLI

To enable retransmission under Rulebase or Service Level base, use the following configuration:

```plaintext
configure
  active-charging service <ecs_service_name>
    rulebase <rulebase name>
      retransmissions-counted
    end
```

Notes:

Use the `no retransmission counted` command to disable retransmission counted feature.

To verify your configuration, in the Exec mode, enter the following command:

`show active-charging rulebase name <rulebase_name>`

Configuring Websockets

This section lists the syntax to enable or disable websocket flow detection feature.

To enable websocket flow detection feature, use the following configuration:

```plaintext
configure
  active-charging service <ecs_service_name>
    rulebase <rulebase name>
      websocket flow-detection <protocol>
    end
```
Notes:
Use the `no websocket flow-detection` command or `default websocket flow-detection` command to disable retransmission counted feature.
To verify your configuration, in the Exec mode, enter the following command:

```
show active-charging rulebase name <rulebase_name>
```

### Configuring URL Filtering Feature

This section describes how to configure the URL Filtering feature to simplify rules for URL detection. To create a group-of-prefixed-URLs, use the following configuration:

```
configure

active-charging service <ecs_service_name>

  group-of-prefixed-urls <prefixed_urls_group_name>

end
```

To configure the URLs to be filtered in the group-of-prefixed-URLs, use the following configuration:

```
configure

active-charging service <ecs_service_name>

  group-of-prefixed-urls <prefixed_urls_group_name>

    prefixed-url <url_1>
    ...
    prefixed-url <url_10>

end
```

To enable or disable the group in the rulebase for processing prefixed URLs, use the following configuration:

```
configure

active-charging service <ecs_service_name>

  rulebase <rulebase_name>

    url-preprocessing bypass group-of-prefixed-urls <prefixed_urls_group_name>
    ...
    url-preprocessing bypass group-of-prefixed-urls <prefixed_urls_group_name>

end
```

Notes:
• A maximum of 64 group-of-prefixed-urls can be created and configured.
• A maximum of 10 prefixed URLs can be configured in each group-of-prefixed-urls.
• In a rulebase, multiple group-of-prefixed-urls can be configured to be filtered.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging group-of-prefixed-urls name <prefixed_urls_group_name>
```

Configuring URL-based Re-addressing

The URL-based re-addressing feature is configured and enabled via system CLI commands using `charging-action` command options within an Active Charging Service.

Use the following commands in the ACS Charging Action Configuration Mode to configure the URL server to re-address for the specified charging action.

```
configure

active-charging service <service_name>

  charging-action <charging_action_name>

    flow action url-readdress server <ipv4_address> [ port <port_number> ]

    no flow action

end
```

Configuring X-Header Insertion and Encryption Feature

This section describes how to configure the X-Header Insertion and Encryption features, collectively known as Header Enrichment.

Configuring X-Header Insertion

This section describes how to configure the X-Header Insertion feature.

❗️ **Important:** This feature is license dependent. Please contact your Cisco account representative for more information.

To configure the X-Header Insertion feature:

**Step 1** Create/configure a ruledef to identify the HTTP packets in which the x-headers must be inserted. For information on how to create/configure ruledefs, see the Configuring Rule Definitions section.

**Step 2** Create/configure a rulebase and configure the charging-action, which will insert the x-header fields into the HTTP packets. For information on how to create/configure rulebases, see the Configuring Rulebase section.
Step 3  Create the x-header format as described in the Creating the x-header Format section.

Step 4  Configure the x-header format as described in the Configuring the x-header Format section.

Step 5  Configure insertion of the x-header fields as described in the Configuring Charging Action for Insertion of X-Header Fields of x-header Fields section.

Creating the X-Header Format

To create an x-header format, use the following configuration:

```
configure

active-charging service <ecs_service_name>

xheader-format <xheader_format_name>

end
```

Configuring the X-Header Format

To configure an x-header format, use the following configuration:

```
configure

active-charging service <ecs_service_name>

xheader-format <xheader_format_name>

insert <xheader_field_name> { string-constant <xheader_field_value> | variable { bearer { 3gpp { apn | charging-characteristics | charging-id | imei | imsi | qos | rat-type | s-mcc-mnc | sgsn-address } | acr | customer-id | ggsn-address | mdn | msisdn-no-cc | radius-calling-station-id | session-id | sn-rulebase | subscriber-ip-address | username } | encrypt | http { host | url } }

end
```

Configuring Charging Action for Insertion of X-Header Fields

To configure a charging action for insertion of x-header fields, use the following configuration:

```
configure

active-charging service <ecs_service_name>

charging-action <charging_action_name>

xheader-insert xheader-format <xheader_format_name> [ encryption rc4md5 [ encrypted ] key <key> ] [ first-request-only ] [ -noconfirm ]

end
```

Notes:
- If rc4md5 encryption is configured in the charging action, it will take precedence over RSA certificate based encryption for flows hitting particular charging action.
Configuring X-Header Encryption

This section describes how to configure the X-Header Encryption feature.

**Important:** This feature is license dependent. Please contact your Cisco account representative for more information.

To configure the X-Header Encryption feature:

**Step 1** Configure X-Header Insertion as described in the Configuring X-Header Insertion section.

**Step 2** Create/configure a rulebase and configure the encryption certificate to use and the re-encryption parameter as described in the Configuring X-Header Encryption section.

**Step 3** Configure the encryption certificate to use as described in the Configuring Encryption Certificate section.

Configuring X-Header Encryption

To configure X-Header Encryption, use the following configuration example:

```
configure

  active-charging-service <ecs_service_name>

  rulebase <rulebase_name>

    xheader-encryption certificate-name <certificate_name>

    xheader-encryption re-encryption period <re-encryption_period>

end
```

Notes:

- This configuration enables X-Header Encryption for all subscribers using the specified rulebase `<rulebase_name>`.
- If the certificate is removed, ECS will continue using the copy that it has. It will only free its copy if the certificate name is removed from the rulebase.
- Changes to x-header format configuration will not trigger re-encryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next re-encryption time to those existing calls for which re-encryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value will not be available, insertion of that parameter will stop.

Configuring Encryption Certificate

To configure the encryption certificate, use the following configuration example:

```
configure

  certificate-name <certificate_name> pem { { data <pem_certificate_data> private-key pem [ encrypted ] data <pem_pvt_key> } | { url <url> private-key pem { [ encrypted ] data <pem_pvt_key> | url <url> } } }
```
Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging xheader-format name <xheader_format_name>
```