Cisco ASR 5000 Series Enhanced Charging Services Administration Guide

Version 12.0

Last Updated April 30, 2011

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

Text Part Number: OL-24898-01

Cisco ASR 5000 Series Enhanced Charging Services Administration Guide

CONTENTS

About this Guide

This document pertains to the features and functionality that run on and/or that are related to the Cisco® ASR 5000 Chassis, formerly the Starent Networks ST40.

Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	Notice Type	Description
	Information Note	Provides information about important features or instructions.
	Caution	Alerts you of potential damage to a program, device, or system.
	Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.
	Electro-Static Discharge (ESD)	Alerts you to take proper grounding precautions before handling a product.

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

Command Syntax Conventions	Description
{ **`keyword`** or *`variable`* }	Required keywords and variables are surrounded by grouped brackets. Required keywords and variables are those components that are required to be entered as part of the command syntax.

Command Syntax Conventions	Description
[**keyword** or *variable*]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by square brackets.
|	With some commands there may be a group of variables from which the user chooses one. These are called alternative variables and are documented by separating each variable with a vertical bar (also known as a pipe filter). Pipe filters can be used in conjunction with required or optional keywords or variables. For example: **{ nonce | timestamp }** OR [**count** *number_of_packets* | **size** *number_of_bytes*]

Contacting Customer Support

Use the information in this section to contact customer support.

For New Customers: 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 is required to this site. Please contact your local sales or service representative for additional information.

For Existing Customers with support contracts through Starent Networks: Refer to the support area of https://support.starentnetworks.com/ for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

Important: For warranty and repair information, please be sure to include the Return Material Authorization (RMA) tracking number on the outside of the package.

Chapter 1
Enhanced Charging Service Overview

This chapter provides an overview of the Enhanced Charging Service (ECS) in-line service, also known as Active Charging Service (ACS).

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 chapter.

This chapter covers the following topics:

Introduction

Cisco's Enhanced Charging Service (ECS) is an in-line service available in the ASR 5000 Multimedia core platforms. It is integrated within the platform, reducing billing-related costs and giving mobile operators the ability to offer tiered, detailed, and itemized billing to 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.

Charging Subsystem

The 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 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)
- 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)
- 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.

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 inspection is examining the IP header (Layer 3) or UDP or TCP header (Layer 4). Deep-packet inspection is the examination of Layer 7, which contains Uniform Resource Identifier (URI) information. Shallow packet analyzers typically determine the destination IP address or port number of a terminating proxy, whereas deep-packet analyzers typically identify the destination of a terminating proxy.

Deep Packet Inspection

In some cases, Layer 3 and 4 analyzers that identify a trigger condition are insufficient for billing purposes, so Layer 7 is used.

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

Supported Accounting and Charging Interfaces

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)

Licensing

ECS is a licensed in-line service feature requiring one of the following licenses to be installed on the chassis:

- Cisco PID [ASR5K-00-CS01ECG1] *Enhanced Charging Bundle 1 1k Sessions*, or Starent Part Number [600-00-7526] *Enhanced Charging Bundle 1 1k Sessions* — To enable and configure ECS functionality.
- Cisco PID [ASR5K-00-CS01ECG2] *Enhanced Charging Bundle 2 1k Sessions*, or Starent Part Number [600-00-7574] *Enhanced Charging Bundle 2 1k Sessions* — To enable and configure Diameter and DCCA functionality with ECS.

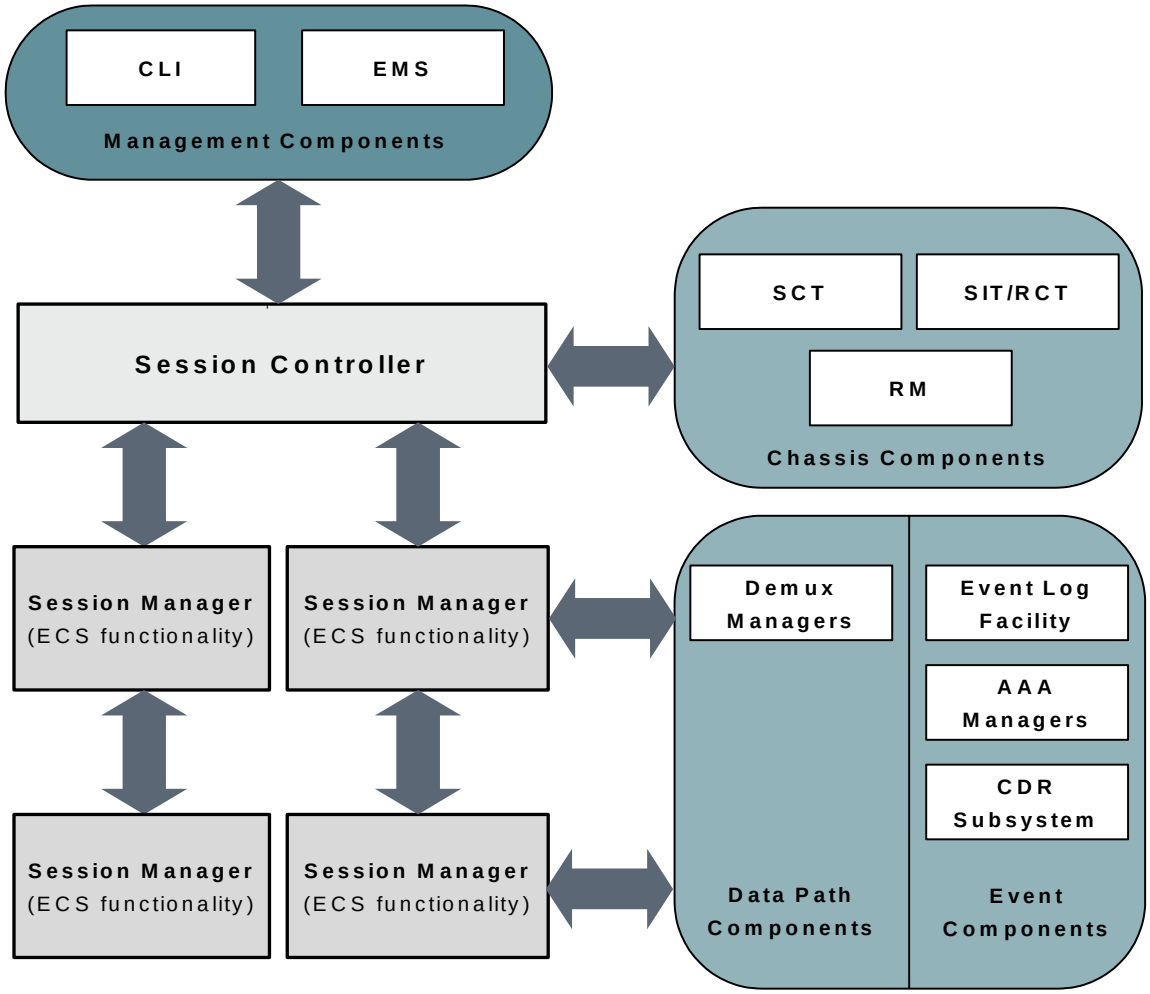
Important: For information on additional licenses required for enhanced or customer-specific features contact your local sales representative.

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

ECS Architecture

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

Figure 1. ECS Architecture



How ECS Works

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

- **Content Service Steering**—Redirects incoming traffic to the ECS subsystem.
- **Protocol Analyzer**—Performs inspection of incoming packets.
- **Rule Definitions**—Specifies the packets to inspect or the charging actions to apply to packets based on content.
- **Rulebases**—Allows grouping one or more number of rule definitions together to define the billing policies for individual subscribers or group of subscribers.

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

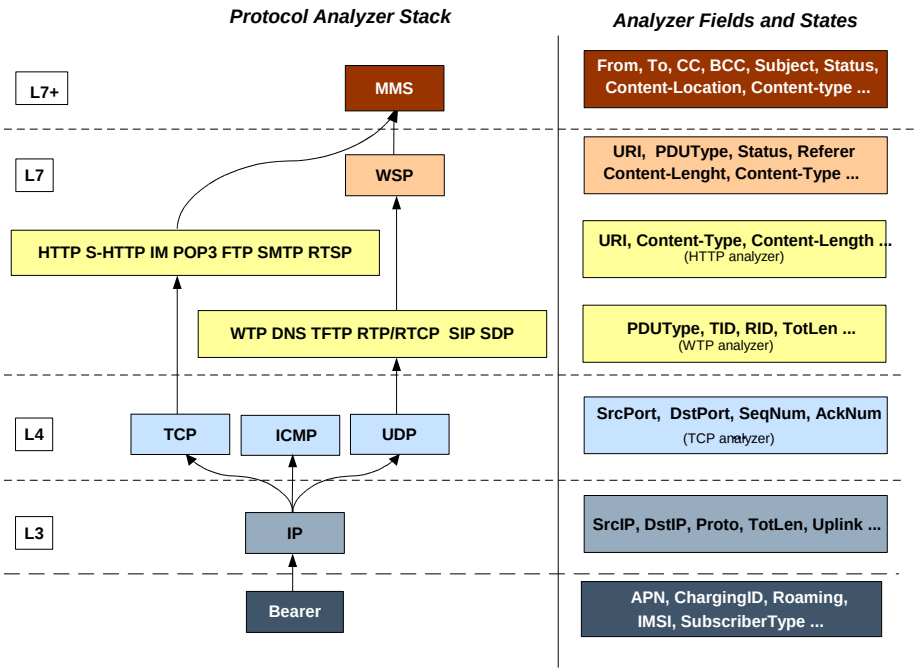
The Protocol Analyzer performs a stateful packet inspection of complex protocols, such as FTP, RTSP, and SIP, which dynamically open ports for the data path, so the payload can be classified according to content.

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 (i.e., 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 2. 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.

Rule Definitions

Rule definitions (ruledefs) are user-defined expressions, based on protocol fields and protocol-states, which define what actions to take when specific field values are true. 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”. Integer type expressions like “packet size” and “sequence number” can be used with comparison operators like “=”, “!=”, “>=”, “<=”. 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 ruledefs can be configured for charging.
- **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 configures 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.

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:

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

–or–

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

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

```
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.

```
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
```

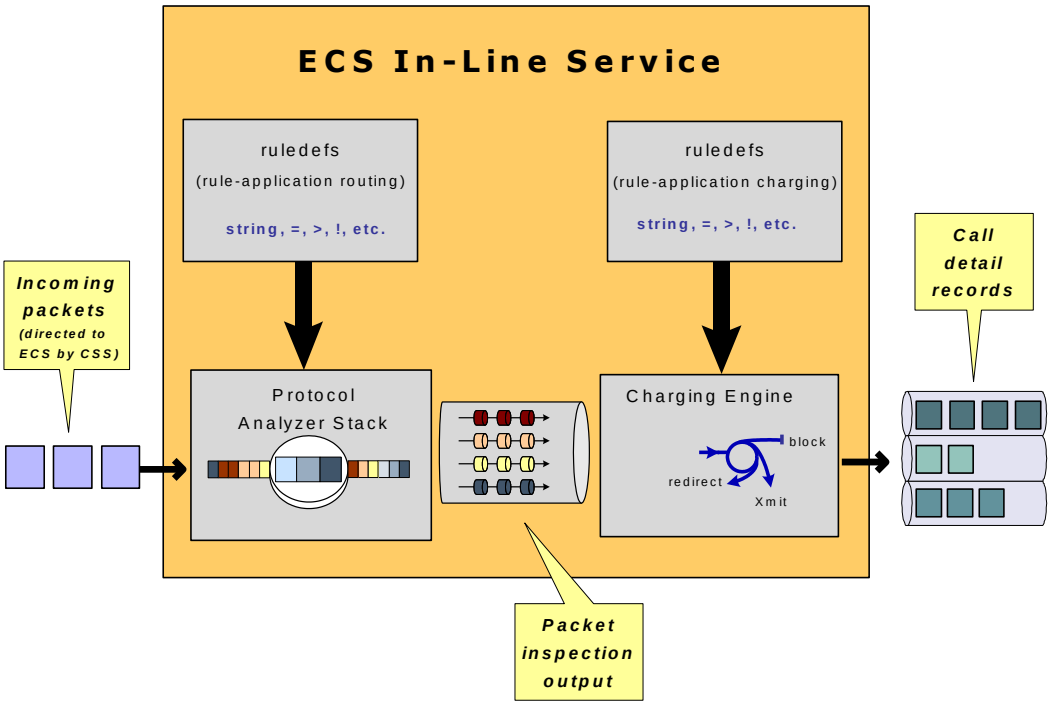
```
    action priority 1000 ruledef catch-all charging-action charge-by-
duration

    [ ... ]

    exit
```

The following figure illustrates how ruledefs interact with the Protocol Analyzer Stack and Action Engine to produce charging records.

Figure 3. ECS In-line Service Processing

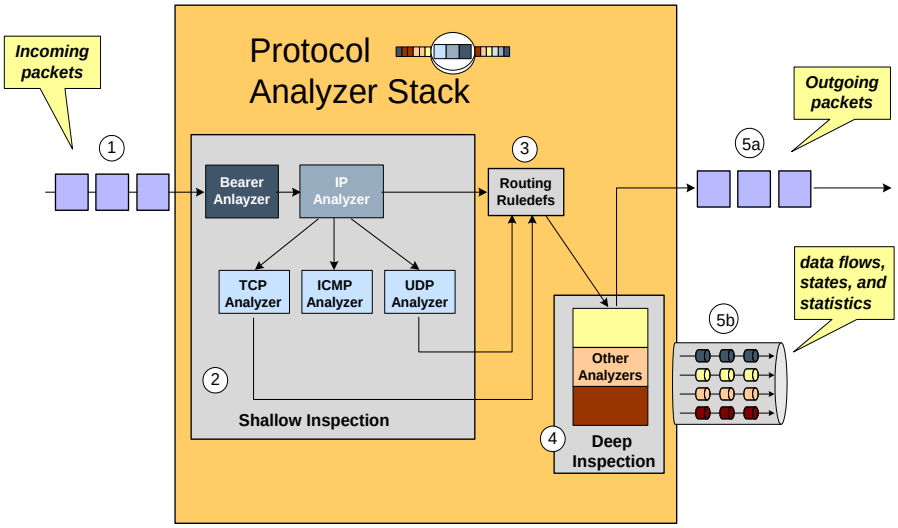


Packets entering the ECS subsystem must first pass through the Protocol Analyzer Stack where routing ruledefs apply to determine which packets to inspect. Then output from this inspection is passed to the Charging Engine, where charging ruledefs apply to perform actions on the output.

Routing Ruledefs and Packet Inspection

The following figure and the steps that follow describe the details of routing ruledef application during packet inspection.

Figure 4. Routing Ruledefs and 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 7or 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.

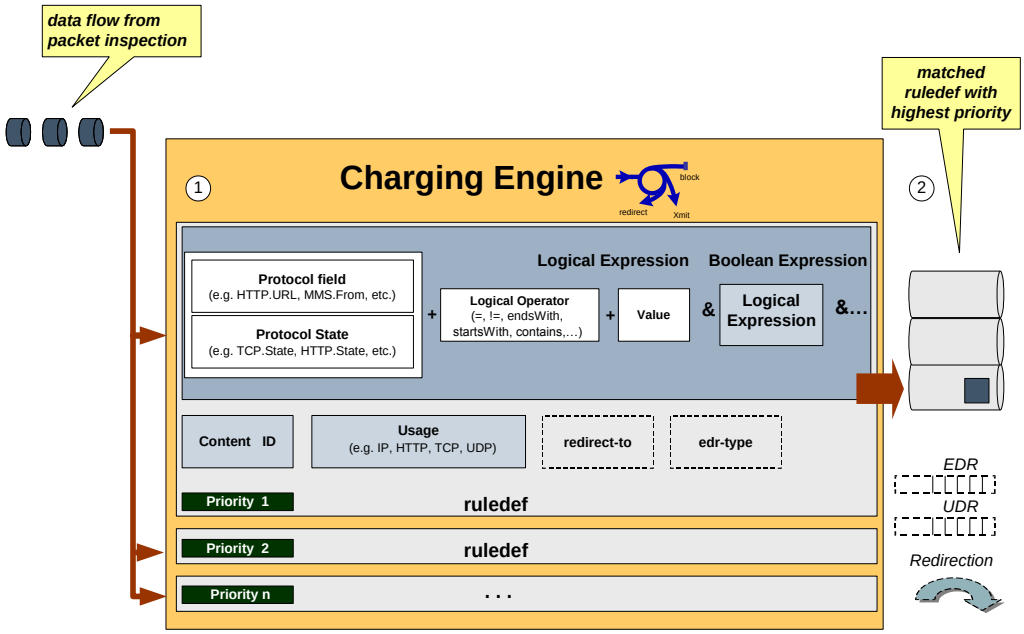
Step b The output of that analysis flows into the Charging Engine, where an action can be applied. Applied actions include redirection, charge value, and billing record emission.

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 5. Charging Ruledefs and Charging Engine



(rule-application charging)

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.

Group-of-Ruledefs

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

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, then the group will be optimized.

When a new ruledef is added, it is checked if it is included in any group-of-ruledefs, and whether it needs to be optimized, and so on.

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 there exists a need to install/remove a large set of predefined rules at the same time. Though this is possible over the Gx interface (with a limit of 256) it requires a lot 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.

Rulebase

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.

Enhanced Services in ECS

This section describes the enhanced and extended features supported in ECS.

Important: Features described in this section are license dependent. If you have not previously purchased licenses for these services, contact your sales representative for more information.

Session Control in ECS

In conjunction with the Cisco ASR 5000 chassis, the ECS provides a high-level network flow and bandwidth control mechanism through 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**—This 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**—This 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 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.

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.

Fair Usage Feature

The Fair Usage feature enables resource management at two levels:

- **Instance-Level Load Balancing**—Enables load balancing of calls based on resource usage for in-line service memory allocations. If an in-line service is configured on the chassis, all resource allocation and release (memory) would involve maintaining instance-level credit usage. Sessions would be more equally distributed based on the in-line memory credits rather than the number of sessions running on individual instances.
- **Subscriber Session Resource Monitoring**—Enables monitoring individual subscriber resource usage and restricting unacceptable usage of resources by subscriber sessions. Every subscriber session will have a free ride until the operator configured threshold is reached. After that, any new resource requirement may be allowed based on the entitled services and the currently available memory in the system. Once resource allocation is failed for a subscriber session, the in-line service application requesting resource manages the failure handling.

  Operators can configure when the monitor action is initiated as a percentage of memory usage. By default, if the feature is enabled, the monitor action would be initiated at 50% threshold. Monitor action includes allowing or failing a particular resource allocation request. Any new resource allocation, once after the threshold is hit, is subject to an evaluation before allowing allocation. The requested resource is compared against the average available memory per session with a configurable per session waiver (default set to 10%). If the instance credit usage goes below a certain configurable percentage of the threshold, monitor action is disabled (default set to 5%) to avoid possible ping pong effect of enabling and disabling monitor action.

  Monitoring memory usage of individual subscriber sessions and providing preferential treatment is based on the services that the subscriber is entitled to. The subscriber session will be entitled to services as configured in the rulebase. Every subscriber session would have free ride until the operator configured threshold is reached. After that, any new resource requirement may be allowed based on the following:

  - Rulebase configured for the subscriber
  - Current available memory in the system

It is recommended that the parameters be configured only after continuous evaluation and fine tuning of the system.

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*.

IP Readdressing Feature

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.

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.

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 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.

X-Header Insertion and Encryption Feature

This section describes the X-Header Insertion and Encryption features, which enables to append subscriber information to HTTP header to be used by end application, such as mobile advertising insertion (MSISDN, IMSI, IP address, user-customizable).

Important: This feature is license dependent. Please contact your local sales representative for more information.

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 an operator wants to insert header field *x-rat-type* in the HTTP header with its value being *rat-type*, i.e. 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:

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.

Limitations to the Header Insertion Feature

The following are limitations to insertion of x-header fields in HTTP headers:

- The packet size is assumed to be less than “Internal MED MTU size, the size of header fields inserted”. Header insertion does not occur after the addition of the fields, if the total length of packet exceeds the internal MTU size.
- 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.
- Window size scaling is not handled in the case of header insertion. Header insertion does not occur if the resulting packet after header insertion exceeds the advertised TCP window size of the server.
- Currently 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 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.

Post Processing Feature

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.

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.

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 rule/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 Active Charging Service Configuration Mode.

A maximum of 10 timedefs can be created in an ECS service.

Step 2 Timedefs are configured in the 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 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. 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

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). I.e., 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.

Important: A prefixed URL can be detected and stripped if it is of the type “http://www.xyz.com/http://www.abc.com”. Here, “http://www.xyz.com” will be stripped off. But in “http://www.xyz.com/www.abc.com”, it cannot detect and strip off “http://www.xyz.com” as it looks for occurrence of “http” or “https” within the URL.

TCP Proxy

The TCP Proxy feature enables the ASR 5000 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, Peer-to-Peer Application Control and Detection, DCCA, and Partial Application Headers features.

TCP Proxy along with other capabilities enables the ASR 5000 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 5000 chassis, the TCP Proxy functionality can be enabled or disabled and configured from the CLI, enabling the ASR 5000 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.

Important: Once TCP Proxy is enabled for a connection, the connection will remain proxied for the lifetime of the connection. TCP Proxy cannot be disabled for the flow.

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, there is only a single TCP connection between subscriber and server for a particular flow. 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 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 Control and Detection 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 following steps, and then enter the Gi side TCP IP Stack of proxy:
     - IP Analysis (support for IP reassembly)
     - Network Address Translation Processing
     - Shallow/Deep Packet TCP Analysis (support for TCP OOO)
     - Stateful Firewall Processing
     - Application Control and Detection Processing
     - DPI Analysis
     - 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.
- 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, i.e. 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, i.e. 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.

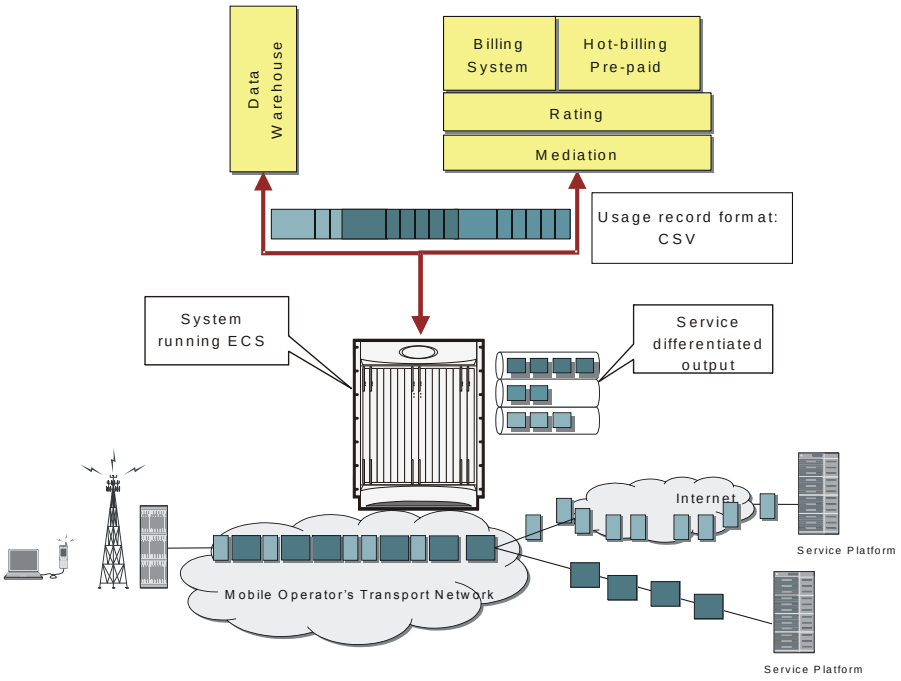
Since TCP state and previous-state rules are now matched based on state on Gi side connection, ECS will not be able to support all the existing use-cases with the existing configuration. New ruledefs based on the new rules (tcp proxy-state and tcp proxy-prev-state) need to be configured to support existing 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, i.e. 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 2 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.

ECS Deployment

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

Figure 6. Deployment of ECS in a Mobile Data Network



Accounting Interfaces

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

Important: Some feature described in this section are license dependent. If you have not previously purchased licenses for these services, contact your sales representative for more information.

GTPP Accounting

Important: GTPP accounting is only available for UMTS networks.

GTTP accounting in ECS allows the collection of counters for different types of data traffic, and including that data in a G-CDR that is sent to a Charging Gateway Function (CGF).

Standard G-CDRs do not have an attribute which defines traffic counters depending upon the traffic type but they do have a field named “Record Extensions” where all vendor-specific information can be included. ECS includes the counters for different types of data traffic in this field when sending a G-CDR.

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 *Gx Interface Support* chapter in the core network service administration guide.

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 plaintext 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 *Gy Interface Support* chapter in the core network service administration guide.

Standard GGSN Call Detail Records (G-CDRs)

Important: G-CDRs are only available in UMTS networks.

G-CDRs are generated according to 3GPP TS 32.251 V6.6.0.

Currently ECS supports generation of CDRs using AAAMgrs only.

G-CDR Format

The G-CDRs can be in ASN.1 format.

Important: For more information on G-CDR fields, refer to the *AAA and GTP Interface Administration and Reference Guide*.

Enhanced GGSN Call Detail Records (eG-CDRs)

Important: eG-CDRs are only available in GGSN networks.

The ECS also supports enhanced G-CDRs, which is an enhanced format of standard G-CDRs to provide greater portability of charging information.

eG-CDRs are compliant with 3GPP TS 32.298 v6.5.0 for Rel. 6 based dictionaries, and with 3GPP TS 32.298 v7.4.0 for Rel. 7 based dictionaries.

By default, the G-CDR does not support the traffic and vendor specific records. To support a traffic and vendor specific record, the ECS must be configured to generate eG-CDRs. eG-CDRs are useful to implement TBC and FBC to ECS.

eG-CDR supports customer specific formats configured in Ga context in a GGSN service with standard or custom specific GTPP dictionaries.

eG-CDR Format

The eG-CDRs can be in ASN.1 format or in delimiter-separated ASCII format. Configuring the eG-CDR encoding type is a CLI-configurable parameter. The default encoding type is ASN.1. When configuring the eG-CDR encoding type as ASCII, the delimiter character can be specified as either ":" (colon), "," (comma), or "|" (pipe). The default delimiter character is "|" (pipe).

Triggers to Update eG-CDRs

The following table lists the trigger conditions to update charging information in an eG-CDR.

Table 1. Triggers for charging information update in eG-CDR

Triggers	Description and Action
PDP context modification	When a change of PDP context conditions (QoS change, SGSN change, PLMN Id change, RAT change) occurs a set of List of Service Data (LOSDV) and List of Traffic Volume (LOTV) containers, i.e. all active service data flow containers, will be added to eG-CDR.A maximum of 10 LOSDV containers are supported per eG-CDR.A maximum of 8 LOTV containers are supported per eG-CDR.
Tariff time change	When a change of tariff time occurs a set of List of Service Data (LOSDV) and List of Traffic Volume (LOTV) containers, i.e. all active service data flow containers, will be added to eG-CDR.A maximum of 10 LOSDV containers are supported per eG-CDR.A maximum of 8 LOTV containers are supported per eG-CDR.
Failure handling procedure triggering	When the failure handling mechanism is triggered and the failure action is set to “continue” a set of List of Service Data (LOSDV) and List of Traffic Volume (LOTV) containers, i.e. all active service data flow containers, will be added to eG-CDR.A maximum of 10 LOSDV containers are supported per eG-CDR.A maximum of 8 LOTV containers are supported per eG-CDR.
Service data flow report	When an expiry of time limit, volume limit or termination is detected for a service data flow a set of List of Service Data (LOSDV) container is added to eG-CDR.A maximum of 10 LOSDV containers are supported per eG-CDR.
CDR closure	When a CDR closure occurs all active List of Service Data (LOSDV) container is added to eG-CDR.A maximum of 10 LOSDV containers are supported per eG-CDR.

Triggers to Close eG-CDRs

The following table lists the trigger conditions to close the eG-CDR.

Table 2. Triggers for closing an eG-CDR

Triggers	Description and action
End of PDP context in GGSN	De-activation of the PDP context in the GGSN closes the eG-CDR. The trigger condition covers: • Termination of PDP context • Any abnormal release of PDP context

Triggers	Description and action
Partial record reasons	eG-cDRs can be closed due to operation parameters and conditions.This trigger reason covers: • Data volume limit • Time duration limit • Maximum number of charging condition changes (QoS/tariff time change) • Management intervention • MS/Subscriber time zone change • Inter PLMN SGSN change • Radio Access Technology (RAT) change

Important: For more information on eG-CDR fields, refer to the *AAA and GTP Interface Administration and Reference Guide*.

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.

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 callline (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:

```
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:

```
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 Record Generation

ECS provides for the generation of charging data files, which can be periodically retrieved from the system and used as input to a billing system for post processing.

The results of traffic analyzer are used to generate Session usage data. The generated usage data are in a standard format, so that the impact on the existing billing system is minimal and at the same time, these records contain all the information required for billing based on the content.

The accounting records also contain the information to identify the user, with Dynamic address assignment and information to obtain the URL for HTTP content request or a file-name or path from FTP request, the type of service from the first packet of the connection, and transaction termination information so that the billing system can decide transaction success or failure.

Charging records support details of the termination, such as which end initiated the termination, termination type, for example RST, FIN, and so on. And, in case of HTTP 1.1, whether or not the connection is still open.

ECS supports pipelining of up to 32 HTTP requests on the same TCP connection. Pipeline overflow requests are not analyzed. Such overflow requests are treated as http-error. The billing system, based on this information, decides to charge or not charge, or refund the subscriber accordingly.

To cover the requirements of standard solutions and at the same time, provide flexible and detailed information on service usage, ECS provides following type of usage records:

- Standard GGSN - Call Detail Records (G-CDRs)
- Enhanced GGSN - Call Detail Records (eG-CDRs)
- Event Detail Records (EDRs)
- Usage Detail Records (UDRs)

EDR/UDR/FDR (xDR) Storage

The system allocates 512 MB of memory on the packet processing card's RAM to store generated charging detail record files (xDRs). The generated xDRs are stored in CSV format in the /records directory on the packet processing card RAM. As this temporary storage space (size configurable) reaches its limits, the system deletes older xDRs to make room for new xDRs. Setting gzip file compression extends the storage capacity approximately 10:1.

Because of the volatile nature of the memory, xDRs can be lost due to overwriting, deletion, or unforeseen events such as power or network failure or unplanned chassis switchover. To avoid loosing charging and network analysis information, configure the CDR subsystem in conjunction with the L-ESS/external storage to offload the xDRs for storage and analysis. Or, configure the system to push records to the L-ESS/external storage.

Hard Disk Support on SMC Card

When using the hard disk for EDR/UDR storage, EDR/UDR files are transferred from RAMFS on the PSC card to the hard disk on the SMC card. The hard disk may also be used to store any data that needs to be backed up.

The secondary SMC card also contains a hard disk which serves as a redundant, and becomes active during an SMC failover. The hard disk on the secondary is mirrored to the hard disk on the primary in order to avoid any data loss. Basically, the drives are raid-1 redundant.

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 GGSN networks.

ECS also supports FBC and TBC methods for postpaid billing. For more information on FBC and TBC in ECS, see the Enhanced Services 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.

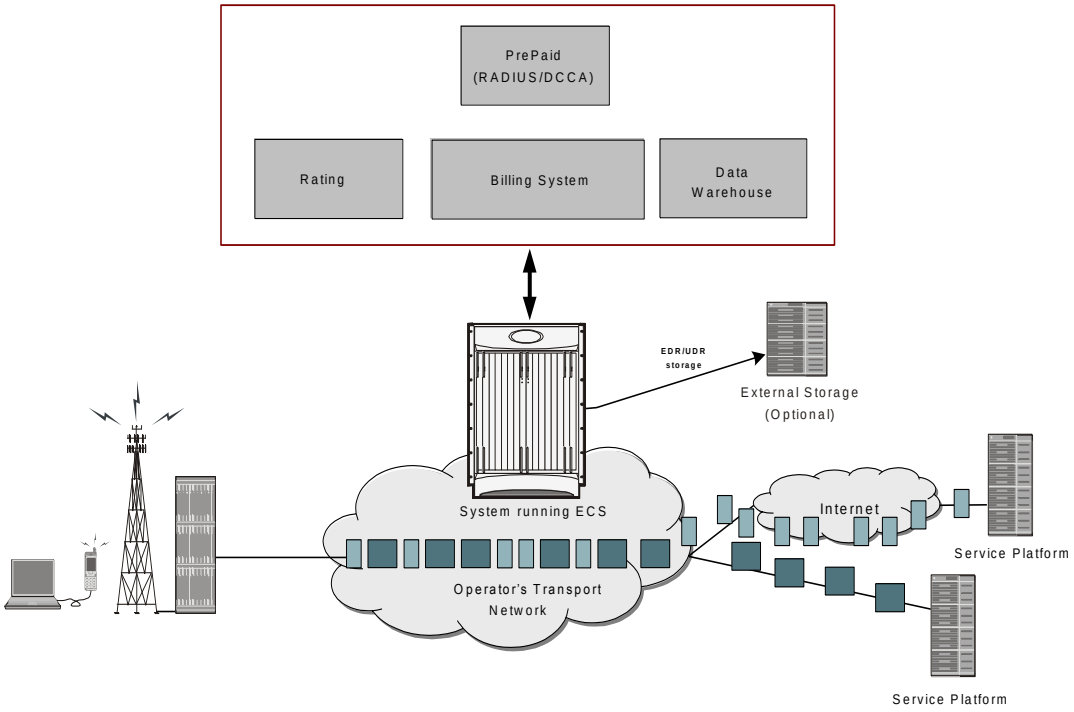
If the user has not used up the purchased credit, the server grants quota and if no credit is available to the subscriber the call will be disconnected. ECS and DCCA manage this functionality by providing the ability to set up quotas for different services.

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 with ECS.

Figure 7. Prepaid Billing Scenario with ECS



Credit Control Application (CCA) in ECS

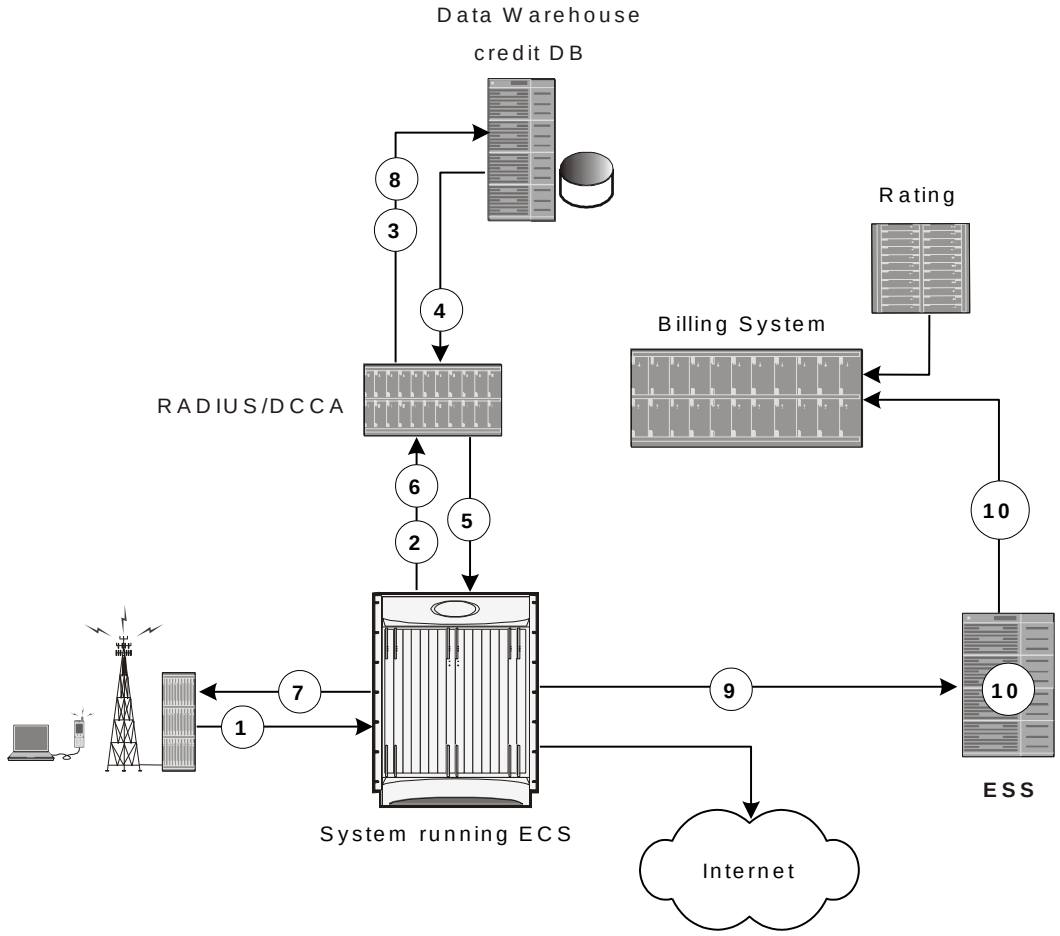
This section describes the credit control application that is used to implement real-time credit-control for a variety of end user services such as network access, Session Initiation Protocol (SIP) services, messaging services, download services, and so on. It provides a general solution to the real-time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

Figure 8. Prepaid Charging in GPRS/UMTS/CDMA-2000 Networks



Step 1 Subscriber session starts.

Step 2 System sends request to CCA for subscriber's quota.

Step 3 CCA sends request to Data Warehouse (DW) credit quota for subscriber.

Step 4 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.

Step 5 CCA sends the amount of quota required to fulfill the subscriber's initial requirement to the system.

Step 6 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.

Step 7 Subscriber session ends after either quota exhausts for subscriber or subscriber terminates the session.

Step 8 CCA returns unused quota to DW for update to subscribers Credit DB.

Step 9 EDRs and UDRs are periodically SFTPd from system memory to the L-ESS/external storage, if deployed or to billing system directly as they are generated. Or, if configured, pushed to the L-ESS/external storage at user-configurable intervals.

Step 10 The L-ESS/external storage periodically sends records to the billing system or charging reporting and analysis system.

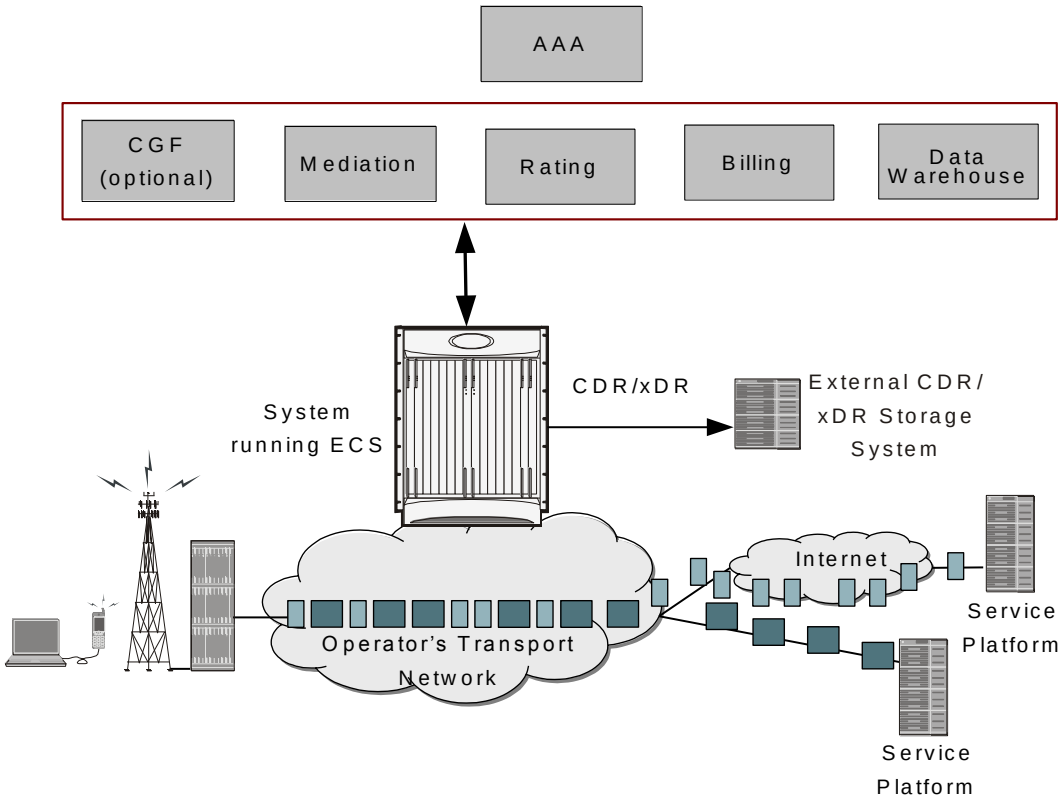
Important: For more information on ESS contact your local sales representative.

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.

Figure 9. Postpaid Billing System Scenario with ECS

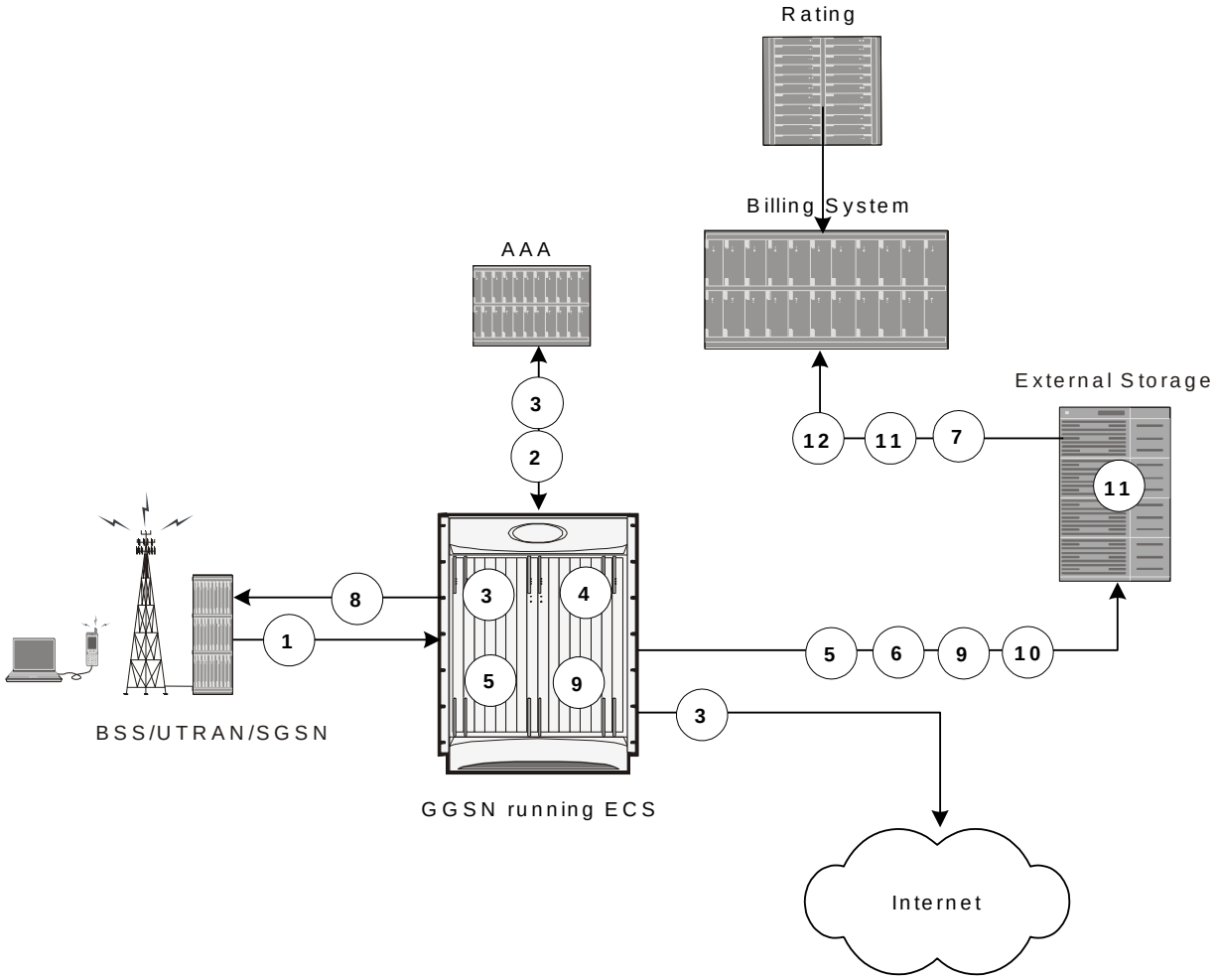


How ECS Postpaid Billing Works

ECS Postpaid Billing in GPRS/UMTS Networks

The following figure and steps describe how ECS works in a GPRS/UMTS network for postpaid billing.

Figure 10. Postpaid Billing with ECS in GPRS/UMTS Network



Step 1 The subscriber initiates the session.

Step 2 After subscriber authentication and authorization, the system starts the session.

Step 3 Data packet flow and accounting starts.

Step 4 System periodically generates xDRs and stores them to the system memory.

Step 5 System generates G-CDRs/eG-CDRs and sends them to billing system as they are generated.

Step 6 The billing system picks up the CDR files periodically.

Step 7 Subscriber session ends after subscriber terminates the session.

Step 8 The system stores the last of the xDRs to the system memory and final xDRs are SFTPd from system memory to L-ESS/external storage, if deployed or to billing system directly.

Step 9 System sends the last of the G-CDRs/eG-CDRs to the billing system.

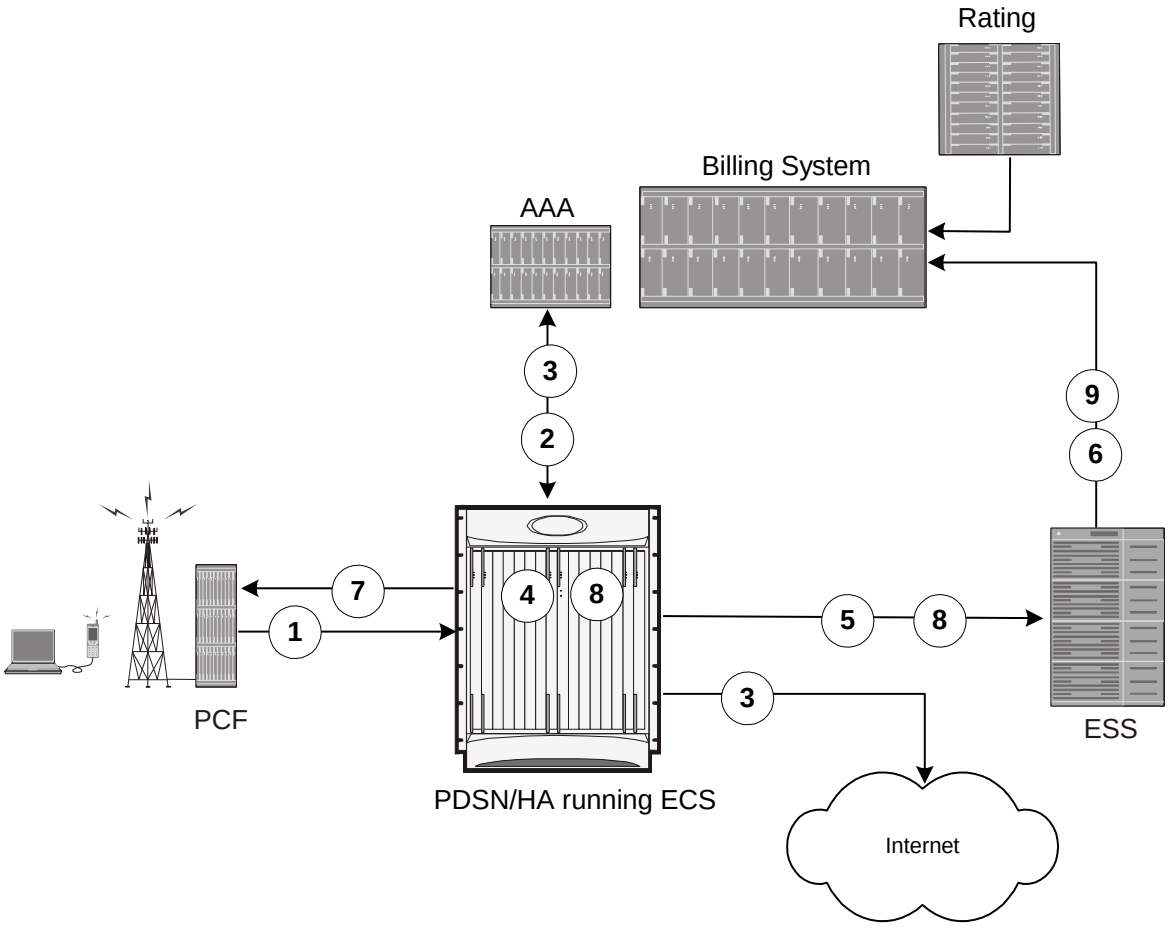
Step 10 File Generation Utility, FileGen in external storage periodically runs to generate G-CDRs/eG-CDRs files for billing system and send them to the billing system.

Step 11 The billing system picks up the xDR files from the L-ESS/external storage periodically.

Postpaid Billing in CDMA-2000 Networks

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

Figure 11. Postpaid Billing with ECS in CDMA-2000 Network



Step 1 The subscriber initiates the session.

Step 2 After subscriber authentication and authorization, the system starts the session.

Step 3 Data packet flow and accounting starts.

Step 4 System periodically generates xDRs and stores them to the system memory.

Step 5 EDRs/UDRs are periodically SFTPd from system memory to L-ESS/external storage, if deployed or to billing system directly as they are generated.

Step 6 The billing system picks up the xDR files from the L-ESS/external storage periodically.

Step 7 Subscriber session ends after subscriber terminates the session.

Step 8 The system stores the last of the xDRs to the system memory and final xDRs are SFTPd from system memory to the L-ESS/external storage, if deployed or to billing system directly.

Step 9 The L-ESS/external storage finally sends xDRs to the billing system.

External Storage System

Important: For information on availability/support for L-ESS, contact your local sales representative.

The Local - External Storage System (L-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 L-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 L-ESS from the chassis, or the chassis can push the xDR files to the L-ESS using SFTP protocol. In the Push mode, the L-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 L-ESS successfully.

The system running with ECS stores xDRs on an L-ESS, and the billing system collects the xDRs form the L-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 L-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.

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.

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.

Impact on xDR File Naming

The xDR file name is limited to 256 characters with 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
- *FileSeqNumber*—Unique file sequence number for the file with 9 digit integer having range from 000000000 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 (i.e. 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

To perform the initial configuration:

Step 1 Create the ECS administrative user account as described in the Creating the ECS Administrative User Account section.

Step 2 Install the ECS license as described in the Installing the ECS License section.

Step 3 Enable enhanced charging as described in the Enabling Active Charging Service section.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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

For ECS functionality one of the following licenses must be installed on the chassis:

- Cisco PID [ASR5K-00-CS01ECG1] *Enhanced Charging Bundle 1 1k Sessions*, or Starent Part Number [600-00-7526] *Enhanced Charging Bundle 1 1k Sessions* — To enable and configure ECS functionality.
- Cisco PID [ASR5K-00-CS01ECG2] *Enhanced Charging Bundle 2 1k Sessions*, or Starent Part Number [600-00-7574] *Enhanced Charging Bundle 2 1k Sessions* — To enable and configure Diameter and DCCA functionality with ECS.

For information on obtaining and installing 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:

- If you are using the Category-based Content Filtering application, refer to the *Content Filtering Services Administration Guide* for specific configuration procedures.

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 Enhanced Charging Service as described in the Creating the Active 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 Rulebases section.

Step 5 Set EDR formats as described in the Setting EDR Formats section.

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

Step 7 Enable charging record retrieval as described in the Enabling Charging Record Retrieval section.

Step 8 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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> [ -noconfirm ]
   end
```

Configuring Rule Definitions

This section describes how to create and configure rule definitions. A ruledef can be a charging, routing, or a post-processing rule. If the same ruledef is to be used for charging in one rulebase and for post-processing rule in another one, then two separate identical ruledefs must be defined.

The following topics are covered in this section:

- Configuring Charging Rule Definitions
- Configuring Routing Rule Definitions
- Configuring Post-processing Rule Definitions

Configuring Charging Rule Definitions

To create and configure a charging ruledef, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      ruledef <ruledef_name>
         <protocol> <expression> <operator> <condition>
         rule-application charging
         end
```

Notes:

- 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*.

Verifying your Configuration

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

```
show active-charging ruledef charging
```

Configuring Routing Rule Definitions

To create and configure a routing ruledef, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      ruledef <ruledef_name>
         <protocol> <expression> <operator> <condition>
```

```
rule-application routing
end
```

Notes:

- The rule-application command specifies the ruledef type. By default, if not specified, the system configures the ruledef as a charging ruledef.
- 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*.
- Up to 10 rule matches can be configured in one ruledef.

Verifying your Configuration

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

```
show active-charging ruledef routing
```

Configuring Post-processing Rule Definitions

To create and configure a post-processing ruledef, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      ruledef <ruledef_name>
         <protocol> <expression> <operator> <condition>
         rule-application post-processing
         end
```

Notes:

- The rule-application command specifies the ruledef type. By default, if not specified, the system configures the ruledef as a charging ruledef.
- 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*.

Verifying your Configuration

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

```
show active-charging ruledef post-processing
```

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 <group_name> [ -noconfirm ]
         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-ruledefs.

Verifying your Configuration

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

```
show active-charging group-of-ruledefs name <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> [ -noconfirm ]
         content-id <content_id>
         retransmissions-counted
         billing-action [ edr <edr_format> [ wait-until-flow-ends ] | egcdr |
exclude-from-udrs | radius ] +
```

```
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:

```
configure
   active-charging service <ecs_service_name>
      charging-action <charging_action_name>
         flow action readdress { server <ip_address> [ port <port_number> ] |
port <port_number> }
         end
```

Configuring Next Hop Address

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

```
configure
   active-charging service <ecs_service_name>
      charging-action <charging_action_name>
         nexthop-forwarding-address <ip_address>
         end
```

Configuring Rulebases

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:

```
configure

   active-charging service <ecs_service_name>

      rulebase <rulebase_name> [ -noconfirm ]

         flow end-condition { { content-filtering | normal-end-signaling |
timeout + } | { { hagr | handoff | session-end } [ flow-overflow ] + } [ url-
blacklisting ] } edr <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 <ruledefs_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>
```

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

   active-charging service <ecs_service_name>

      edr-format <edr_format_name> [ -noconfirm ]

         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> [ -noconfirm ]
         attribute <attribute> { [ format { MM/DD/YY-HH:MM:SS | MM/DD/YYYY-HH:MM:SS | YYYY/MM/DD-HH:MM:SS | YYYYMMDDHHMMSS | 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
   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
   context <context_name>
      subscriber name <subscriber_name> [ -noconfirm ]
         active-charging rulebase <rulebase_name>
         end
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring a Rulebase in 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:

```
configure
   context <context_name>
      apn <apn_name> [ -noconfirm ]
         active-charging rulebase <rulebase_name>
         end
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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:

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

Notes:

- In 11.0 and later releases, the **medium** option is not supported, and the **medium** keyword 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.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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 following license on the chassis: Cisco PID [ASR5K-00-CS01ECG2] *Enhanced Charging Bundle 2 1k Sessions*, or Starent Part Number [600-00-7574] *Enhanced Charging Bundle 2 1k Sessions*.

Important: Before configuring Diameter or RADIUS CCA, you must configure AAA parameters as described in 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

Step 3 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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:

```
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.

  flow action redirect-url *<redirect_url>* **[clear-quota-retry-timer]**

  The following sample 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.subscriber-
id#&host=#http.host#&httpuri=#http.uri#" clear-quota-retry-timer
   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-id <content_id> ] }

        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> ]

         failure-handling { ccfh-session-timeout <session_timeout>
| { initial-request | terminate-request | update-request } {
continue [ go-offline-after-tx-expiry | retry-after-tx-expiry ] |
retry-and-terminate [ retry-after-tx-expiry ] | terminate }

         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 following license on the chassis: Cisco PID [ASR5K-00-CS01ECG2] *Enhanced Charging Bundle 2 1k Sessions*, or Starent Part Number [600-00-7574] *Enhanced Charging Bundle 2 1k Sessions*.

Important: It is assumed that you have already fully configured the AAA parameters as described in the *AAA Interface Administration and Reference*, and Credit Control Application as described in Configuring Prepaid Credit Control Application (CCA) section for Diameter mode.

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

```
configure
   active-charging service <ecs_service_name>
      credit-control [ group <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> ] ] [ imsi-based { [ prefix
| suffix ] <imsi/prefix/suffix_start_value> } [ to
<imsi/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
<ruledefs_group_name> | ruledef <ruledef_name> } charging-action
<charging_action_name> [ monitoring-key <monitoring_key> ] [
description <description> ] }

         end
```

- *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>* **} + }**

- If the Diameter endpoint parameters are not yet configured, see the *Configuring Diameter Endpoint* section in 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:

```
configure
   context <context_name>
      apn <apn_name>
         dcca peer-select peer <host_name> [ [ realm <realm_name> ] [ secondary-peer <host_name> [ realm <realm_name> ] ] ]
         end
```

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 as described in the *AAA Interface Administration and Reference*, and Credit Control Application as described in Configuring Prepaid Credit Control Application (CCA) section for RADIUS mode.

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

```
configure
   active-charging service <ecs_service_name>
      credit-control [ group <group_name> ]
         mode radius
         exit
      rulebase <rulebase_name>
         cca radius charging context <vpn_context> [ group <group_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:

  cca radius accounting interval *<interval>*
- *Optional:* To specify the user password for RADIUS prepaid services, in the Rulebase Configuration Mode, enter the following command:

  cca radius user-password [encrypted] password *<password>*
- If RADIUS server parameters are not yet configured, configure them as described in the *Configuring AAA Functionality* section of the *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 Creating an ECS ACL section.

Step 2 Apply an ACL to an individual subscriber as described in the Applying an ACL to an Individual Subscriber section.

Step 3 Apply an ACL to the subscriber named default as described in the Applying an ACL to the Subscriber Named default section.

Step 4 Apply the ACL to an APN as described in the Applying the ACL to an APN section.

Step 5 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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 an ECS ACL

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

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

Notes:

- *<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, refer to the *AAA and GTP 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:

Important: This information is only applicable to GGSN networks.

```
configure
   context <context_name>
      apn <apn_name>
```

```
ip access-group <acl_name> [ in | out ]
end
```

Configuring GTPP Accounting

This section describes how to configure GTPP accounting which generates G-CDRs for ECS.

Important: This section assumes that a GGSN service is already fully configured. The GGSN service must be configured in the source context.

Important: If the RADIUS accounting is configured you do not need to configure GTPP accounting.

To configure GTPP accounting, use the following configuration:

```
configure
   gtpp single-source
   context <context_name>
      apn <apn_name>
         accounting-mode none
         exit
      gtpp charging-agent address <ip_address>
      gtpp server <ip_address>
      end
```

Notes:

- The **gtpp single-source** command must be entered before any other configuration commands. If you add it to an existing configuration, make sure that it is the first command implemented after cards and ports are configured. This ensures that this command is implemented before any AAA Manager or Session Manager processes are started.
- The GGSN has a concept of an accounting context. By default the accounting context is the context in which the GGSN service is configured. Make sure that you configure the GTPP charging agent and the GTPP accounting servers.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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

         cdr [ [ push-interval <interval> ] [ push-trigger space-usage-percent
<trigger_percentage> ] [ remove-file-after-transfer ] [ transfer-mode { pull |
push primary { encrypted-url <encrypted_url> | url <url> } [ secondary {
encrypted-secondary-url <encrypted_secondary_url> | 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 { omit
| padded | padded-six-length | unpadded } ] [ storage-limit <limit> ] [ 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 { 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.
- 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 Fair Usage Feature

This section describes how to configure the Fair Usage feature to perform instance level load balancing and subscriber resource usage control.

To configure the Fair Usage feature, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      fair-usage
      fair-usage threshold-percent <usage_threshold>
      fair-usage deact-margin <deactivate_margin>
      fair-usage adjust-factor <adjust_factor>
      fair-usage inline-memory-share <max_mem_for_in-line>
      rulebase <rulebase_name>
         fair-usage session-waiver-percent <session_waiver>
         end
```

Notes:

- **fair-usage** command enables the Fair Usage feature.
- **fair-usage threshold-percent** *<usage_threshold>* command configures when to enable resource monitoring. As long as the amount of available memory is greater than the configured threshold, any memory requests are granted.

  <usage_threshold> is a percent value, and must be an integer from 1 through 100.
- **fair-usage deact-margin** *<deactivate_margin>* command configures when to disable resource monitoring. It is the window size between restricting/not restricting memory utilization.

  For example, if the “fair-usage threshold-percent” is set to 75%, on reaching this threshold resource monitoring is enabled. If the “fair-usage deact-margin” is set to 5%, when memory utilization falls 5% below “fair-usage threshold-percent”, i.e. to 70%, resource monitoring is disabled.

  <deactivate_margin> is a percentage value, and must be an integer from 1 through 100. By default, it is set to 10 percent.
- **fair-usage adjust-factor** *<adjust_factor>* is a hidden CLI command available to operators. This command configures the accuracy of memory reporting. When Session Manager reports its memory consumption, this percentage is added to the value being reported. With this, the reported memory will change as the sessions obtain/release memory.

  <adjust_factor> is a percentage value, and must be an integer from 1 through 100. By default, it is set to 10 percent.

- **fair-usage inline-memory-share** *<max_mem_for_in-line>* is a hidden CLI command available to operators for fine tuning the the performance of this feature. This command configures the amount of memory possibly allocated to in-line services on a Session Manager instance. This limit is per Session Manager.

  <max_mem_for_in-line> is a percentage value, and must be an integer from 1 through 100. By default, it is set to 60 percent.

- **fair-usage session-waiver-percent** *<session_waiver>* command configured in the rulebase configures a waiver for subscribers using the rulebase to use more than the average amount of memory limit configured in the **fair-usage threshold-percent** *<usage_threshold>* command.

  <session_waiver> is a percentage value, and must be an integer from 0 through 1000. By default, it is set to 20 percent.

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 <ruledefs_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.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.
- 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 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:

```
configure
   active-charging service <ecs_service_name>
      rulebase <rulebase_name>
         tcp proxy-mode { dynamic { all | content-filtering | dcca | ip-
readdressing | nexthop-readdressing | xheader-insert + } | static [ port [
<port_number> [ to <port_number> ] ] ] }
         end
```

Verifying your Configuration

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

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

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> [ -noconfirm ]

         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 <group_name> | ruledef <ruledef_name> } charging-action
<charging_action_name> [ description <description> ]

         ...

         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 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 <group_name> [ -noconfirm ]
      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 <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 <group_name>
         ...
         url-preprocessing bypass group-of-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 *<group_name>*

Configuring X-Header Insertion and Encryption Feature

This section describes how to configure the X-Header Insertion and Encryption feature.

Configuring X-Header Insertion

This section describes how to configure the X-Header Insertion feature.

Important: This feature is license dependent. Please contact your sales 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 Charging 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 Rulebases 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.

Step 6 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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> [ -noconfirm ]
      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> [ -noconfirm ]
         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 | 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> [ -noconfirm ]
         xheader-insert xheader-format <xheader_format_name> [ -noconfirm ]
         end
```

Notes:

- The **encrypt** option specifies encryption of x-header field configuration. This option must be configured in the case the X-Header Encryption feature will be configured.

Configuring X-Header Encryption

This section describes how to configure the X-Header Encryption feature.

Important: This feature is license dependent. Please contact your sales 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.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

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> } }
   end
```

Verifying your Configuration

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

```
show active-charging xheader-format name <xheader_format_name>
```

Chapter 3
Verifying and Saving Your Configuration

This chapter describes how to save your system configuration.

Verifying the Configuration

You can use a number of commands to verify the configuration of your feature, service, or system. Many are hierarchical in their implementation and some are specific to portions of, or specific lines in, the configuration file.

Feature Configuration

In many configurations, you have to set and verify specific features. Examples include APN and IP address pool configuration. Using these examples, enter the following commands to verify proper feature configuration:

show apn all

The output displays the complete configuration for the APN. In this example, an APN called apn1 is configured.

```
access point name (APN): apn

authentication context: A

pdp type: ipv4

ehrpd access: N/A

Selection Mode: subscribed

ip source violation: Checked drop limit: 10

ip source violation no accounting: Disabled

accounting mode: gtpp No early PDUs: Disabled

no-interims: Disabled

Bearer Control Mode: none

max-primary-pdp-contexts: 4000000 total-pdp-contexts: 4000000

primary contexts: not available total contexts: not available

max secondary contexts per-subscriber:10 IMS Authorization : disabled

Credit Control : disabled

mbms bearer absolute timeout : 0 mbms bearer idle timeout : 0

mbms ue absolute timeout : 0

permission :

local ip: 0.0.0.0 nexthop gateway addr:

primary dns: 0.0.0.0 secondary dns: 0.0.0.0
```

```
primary nbns: 0.0.0.0 secondary nbns: 0.0.0.0
ppp keep alive period : 0 ppp mtu : 1500
absolute timeout : 0 idle timeout : 0
idle-timeout-activity ignore-downlink: Disabled
long duration timeout: 0 long dur inactivity time: 0
long duration action: Detection
wimax header compression/suppression: none
ip header compression: vj
ip hide service address: Disabled
ip output access-group: ip input access-group:
ipv6 output access-group: ipv6 input access-group:
policy-group in: policy-group out:
permit ip multicast: False
ppp authentication:
eap authentication initial-access-request: authenticate-authorize
allow noauthentication: Enabled imsi authentication: Disabled
msisdn authentication: Disabled
ip destination context : A
Rule Base : default
FW-and-NAT Policy: default
Bandwidth-Policy: default
Link-Monitoring: OFF
Content-Filtering Policy-Id : Not configured
mediation accounting: Disabled
mediation-device context: Not set mediation no early PDUs: Disabled
mediation no-interims: Disabled mediation delay-GTP-response: Disabled
outbound username: N/A
ip address pools: N/A
ip address secondary pools: N/A
```

```
access-link ip-frag: df-ignore
ignore DF-bit data-tunnel: On
ip allocation type: local pool allow user specified ip addr: true
prefer dhcp options: false
allow deferred: false
3gpp qos to dscp mapping:
 qci 1: ef
 qci 2: ef
 qci 3: af11
 qci 4: af11
 qci 5: ef
 qci 6: ef
 qci 7: af21
 qci 8: af21
 qci 9: be
3GPP Qos to DSCP Mapping based on Alloc. Prio:
 qci 5 ( Alloc.P 1): ef
 qci 5 ( Alloc.P 2): ef
 qci 5 ( Alloc.P 3): ef
 qci 6 ( Alloc.P 1): ef
 qci 6 ( Alloc.P 2): ef
 qci 6 ( Alloc.P 3): ef
 qci 7 ( Alloc.P 1): af21
 qci 7 ( Alloc.P 2): af21
 qci 7 ( Alloc.P 3): af21
 qci 8 ( Alloc.P 1): af21
 qci 8 ( Alloc.P 2): af21
 qci 8 ( Alloc.P 3): af21
Copy user-datagram IP TOS : Disabled
```

```
APN defined Charging Characteristics:
 Home Subscribers:
 Behavior Bits: 0x0 Profile Value: 8
 Visiting Subscribers:
 Behavior Bits: 0x0 Profile Value: 8
 Roaming Subscribers:
 Behavior Bits: 0x0 Profile Value: 8
 All (Home/Visiting/Roaming) Subscribers:
 Behavior Bits: 0x0 Profile Value: 8
Subscribers to use APN defined charging characteristics: none
Subscribers to use RADIUS returned charging characteristics: No
PDG Subscribers to use APN defined charging characteristics: none
dhcp service name : Not set
dhcp context name : Not set
dhcp lease expiry policy: auto renew
mobile-ip: Disabled
mobile-ip home-agent : 0.0.0.0
mobile-ip alternate-home-agent(s) : n/a
mobile-ip reverse-tunnel : Enabled
mobile-ip mn-aaa-removal-indication : Disabled
mobile-ip mn-ha-spi: None
mobile-ip mn-ha-hash-algorithm: hmac-md5
proxy-mip: Disabled
proxy-mipv6: Disabled
proxy-mip null-username static home address: Disabled
Tunnel peer load-balancing : random
L3-to-L2 tunnel address-policy no-alloc-validate
tunnel address-policy alloc-validate
NPU QoS Traffic Priority: Derive from packet DSCP
```

```
APN QoS Attributes
 Newcall Policy: Accept
 SDU Error Ratio: Residual BER:
qci 1
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate(in bps): 256000000 Committed Data Rate(in bps): 256000000
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): 65535 Guaranteed Burst Size(bytes): 65535
 Exceed Action: lower-ip-precedence Exceed Action: lower-ip-precedence
 Violate Action: drop Violate Action: drop
qci 2
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate(in bps): 256000000 Committed Data Rate(in bps): 256000000
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): 65535 Guaranteed Burst Size(bytes): 65535
 Exceed Action: lower-ip-precedence Exceed Action: lower-ip-precedence
 Violate Action: drop Violate Action: drop
qci 3
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
```

```
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate(in bps): 256000000 Committed Data Rate(in bps): 256000000
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): 65535 Guaranteed Burst Size(bytes): 65535
 Exceed Action: lower-ip-precedence Exceed Action: lower-ip-precedence
 Violate Action: drop Violate Action: drop
qci 4
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate(in bps): 256000000 Committed Data Rate(in bps): 256000000
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): 65535 Guaranteed Burst Size(bytes): 65535
 Exceed Action: lower-ip-precedence Exceed Action: lower-ip-precedence
 Violate Action: drop Violate Action: drop
qci 5
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate: n/a Committed Data Rate: n/a
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
```

```
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): n/a Guaranteed Burst Size(bytes): n/a
 Exceed Action: n/a Exceed Action: n/a
 Violate Action: drop Violate Action: drop
qci 6
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate: n/a Committed Data Rate: n/a
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): n/a Guaranteed Burst Size(bytes): n/a
 Exceed Action: n/a Exceed Action: n/a
 Violate Action: drop Violate Action: drop
qci 7
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate: n/a Committed Data Rate: n/a
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): n/a Guaranteed Burst Size(bytes): n/a
 Exceed Action: n/a Exceed Action: n/a
 Violate Action: drop Violate Action: drop
```

```
qci 8
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate: n/a Committed Data Rate: n/a
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): n/a Guaranteed Burst Size(bytes): n/a
 Exceed Action: n/a Exceed Action: n/a
 Violate Action: drop Violate Action: drop
qci 9
 Downlink Negotiate Limit: Disabled Uplink Negotiate Limit: Disabled
 Peak Data Rate(in bps): 256000000 Peak Data Rate(in bps): 256000000
 Committed Data Rate: n/a Committed Data Rate: n/a
 Downlink Rate Limit: Disabled Uplink Rate Limit: Disabled
 Burst Size: Burst Size:
 Auto Readjust: Disabled Auto Readjust: Disabled
 Auto Readjust Duration: n/a Auto Readjust Duration: n/a
 Peak Burst Size(bytes): 65535 Peak Burst Size(bytes): 65535
 Guaranteed Burst Size(bytes): n/a Guaranteed Burst Size(bytes): n/a
 Exceed Action: n/a Exceed Action: n/a
 Violate Action: drop Violate Action: drop
APN-AMBR
 Downlink Apn Ambr : Disabled Uplink Apn Ambr : Disabled
 Burst Size: Burst Size:
 Auto Readjust: Enabled Auto Readjust: Enabled
 Auto Readjust Duration: 1 Auto Readjust Duration: 1
```

```
 Violate Action: transmit Violate Action: transmit
ppp accept peer ipv6 ifid : no
ipv6 init router advt interval : 3000
ipv6 init router number of advts : 3
ipv6 address prefix : <none>ipv6 address prefix pool : <none>
ipv6 interface id : <none>
ipv6 dns primary server : <none>
ipv6 dns secondary server : <none>
p-cscf primary ip : 0.0.0.0 p-cscf secondary ip : 0.0.0.0
p-cscf primary ipv6 : <none>
p-cscf secondary ipv6 : <none>
ipv6 egress address filtering : no
ipv6 dsn proxy : no
ipv6 minimum link MTU : 1280
Radius Group: default
Radius Secondary Group: <none>
External AAA Group: default
External AAA Context: A
Radius Returned Framed IP Address 255.255.255.255 Policy: Reject-Call-When-MS-IP
-Not-Supplied
Access-flow traffic-validation: disabled
Virtual APN Configuration:
 None
IPv6 Configuration
 IPv6 initial number of router advertisements: 3
 IPv6 initial router advertisements interval: 3000ms
 IPv6 Prefix Pool: Not defined
 IPv6 Egress address filtering: Disabled
 IPv6 Primary DNS server address: ::
```

```
 IPv6 Secondary DNS server address: ::

GTPP Group : <none> GTPP Accounting Context : <none>

Mobile IPv6 Tunnel MTU : 1500

Mobile IPv6 Tunnel MTU Exceed Action : notify-sender

Mobile IPv6 Home Agent: none

Mobile IPv6 Home Link Prefix: ::/0

Mobile IPv6 Home Address: none
```

Enter the following command to display the IP address pool configuration:

show ip pool

The output from this command should look similar to the sample shown below. In this example, all IP pools were configured in the *isp1* context.

```
context : isp1:

+-----Type: (P) - Public (R) - Private

 | (S) - Static (E) - Resource

 |

 |+----State: (G) - Good (D) - Pending Delete (R)-Resizing

 ||

 ||++--Priority: 0..10 (Highest (0) .. Lowest (10))

 ||||

 ||||+-Busyout: (B) - Busyout configured

 |||| |||||| vvvvv Pool Name Start Address Mask/End Address Used Avail

 ----- --------- --------------- --------------- -------- --------

PG00 ipsec 12.12.12.0 255.255.255.0 0 254 PG00

pool1 10.10.0.0 255.255.0.0 0 65534 SG00

vpnpool 192.168.1.250 192.168.1.254 0 5 Total Pool Count: 5
```

Important: To configure features on the system, use the *show* commands specifically for these features. Refer to the *Cisco Systems ASR 5000 Command Line Interface Reference* for more information.

Service Configuration

Verify that your service was created and configured properly by entering the following command:

show *<service_type> <service_name>*

The output is a concise listing of the service parameter settings similar to the sample displayed below. In this example, a P-GW service called pgw is configured.

```
Service name : pgw1

Service-Id : 1

Context : test1

Status : STARTED

Restart Counter : 8

EGTP Service : egtp1

LMA Service : Not defined

Session-Delete-Delay Timer : Enabled

Session-Delete-Delay timeout : 10000(msecs)

PLMN ID List : MCC: 100, MNC: 99

Newcall Policy : None
```

Context Configuration

Verify that your context was created and configured properly by entering the following command:

show context name *<name>*

The output shows the active context. Its ID is similar to the sample displayed below. In this example, a context named *test1* is configured.

```
Context Name     ContextID     State

------------     ---------     -----

test1            2          Active
```

System Configuration

Verify that your entire configuration file was created and configured properly by entering the following command:

show configuration

This command displays the entire configuration including the context and service configurations defined above.

Finding Configuration Errors

Identify errors in your configuration file by entering the following command:

show configuration errors

This command displays errors it finds within the configuration. For example, if you have created a service named "service1", but entered it as "srv1" in another part of the configuration, the system displays this error.

You must refine this command to specify particular sections of the configuration. Add the **section** keyword and choose a section from the help menu:

show configuration errors section ggsn-service

or

show configuration errors section aaa-config

If the configuration contains no errors, an output similar to the following is displayed:

```
##############################################################################

Displaying Global
AAA-configuration errors
##############################################################################

Total 0 error(s) in this section !
```

Saving the Configuration

Save system configuration information to a file locally or to a remote node on the network. You can use this configuration file on any other systems that require the same configuration.

Files saved locally can be stored in the SMC's CompactFlash or on an installed PCMCIA memory card on the SMC. Files that are saved to a remote network node can be transmitted through FTP, or TFTP.

Saving the Configuration on the Chassis

These instructions assume that you are at the root prompt for the Exec mode:

```
[local]host_name#
```

To save your current configuration, enter the following command:

save configuration *url* **[-redundant] [-noconfirm] [showsecrets] [verbose]**

Keyword/Variable	Description
url	Specifies the path and name to which the configuration file is to be stored. *url* may refer to a local or a remote file. *url* must be entered using one of the following formats: • **{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **file:/{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **tftp://**{ *ipaddress*|*host_name*[*:port#*] } [*/directory*] */file_name* • **ftp://**[*username*[*:pwd*] *@*] { *ipaddress*|*host_name* } [*:port#*] [*/directory*] */file_name* • **sftp://**[*username*[:*pwd*] *@*] { *ipaddress*|*host_name* } [*:port#*] [*/directory*] */file_name* **/flash** corresponds to the CompactFlash on the SMC. **/pcmcia1** corresponds to PCMCIA slot 1. **/pcmcia2** corresponds to PCMCIA slot 2. *ipaddress* is the IP address of the network server. *host_name* is the network server's *hostname*. *port#* is the network server's logical port number. Defaults are: • tftp: 69 - data • ftp: 20 - data, 21 - control • sftp: 115 - data Note: *host_name* can only be used if the **networkconfig** parameter is configured for DHCP and the DHCP server returns a valid nameserv er.dx *username* is the username required to gain access to the server if necessary. *password* is the password for the specified username if required. */directory* specifies the directory where the file is located if one exists. */file_name* specifies the name of the configuration file to be saved. Note: Configuration files should be named with a .cfg extension.
-redundant	Optional: This keyword directs the system to save the CLI configuration file to the local device, defined by the url variable, and then automatically copy that same file to the like device on the Standby SMC, if available. Note: This keyword will only work for like local devices that are located on both the active and standby SMCs. For example, if you save the file to the /pcmcia1 device on the active SMC, that same type of device (a PC-Card in Slot 1 of the standby SMC) must be available. Otherwise, a failure message is displayed. Note: If saving the file to an external network (non-local) device, the system disregards this keyword.
-noconfirm	Optional: Indicates that no confirmation is to be given prior to saving the configuration information to the specified filename (if one was specified) or to the currently active configuration file (if none was specified).

Keyword/Variable	Description
showsecrets	Optional: This keyword saves the CLI configuration file with all passwords in plain text, rather than their default encrypted format.
verbose	Optional: Specifies to display every parameter that is being saved to the new configuration file.

Important: The **`-redundant`** keyword is only applicable when saving a configuration file to local devices. This command does not synchronize the local file system. If you have added, modified, or deleted other files or directories to or from a local device for the active SMC, you must synchronize the local file system on both SMCs.

To save a configuration file called system.cfg to a directory that was previously created called cfgfiles on the SMC's CompactFlash, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```

To save a configuration file called simple_ip.cfg to a directory called host_name_configs, using an FTP server with an IP address of 192.168.34.156, on which you have an account with a username of administrator and a password of secure, use the following command:

```
save configuration
ftp://administrator:secure@192.168.34.156/host_name_configs/
simple_ip.cfg
```

To save a configuration file called init_config.cfg to the root directory of a TFTP server with a hostname of config_server, enter the following command:

`save configuration` *`tftp://config_server/init_config.cfg`*

Chapter 4
Enhanced Charging Service Sample Configuration

The following is a sample configuration for basic ECS functionality.

```
configure
         require active-charging
         active-charging service acs_1
            ruledef rr_http_80
               tcp either-port = 80
               rule-application routing
               exit
            ruledef rr_mms_http_ct
               http content type = application/vnd.wap.mms-message
               rule-application routing
               exit
            ruledef rr_mms_http_url
               http url ends-with .mms
               rule-application routing
               exit
            ruledef rr_mms_wsp_ct
               wsp content type = application/vnd.wap.mms-message
               rule-application routing
               exit
            ruledef rr_mms_wsp_url
               wsp url ends-with .mms
               rule-application routing
```

```
   exit
ruledef rr_wsp_cl_dst_port
   udp dst-port = 9200
   rule-application routing
   exit
ruledef rr_wsp_cl_src_port
   udp src-port = 9200
   rule-application routing
   exit
ruledef rr_wsp_co_dst_port
   udp dst-port = 9201
   rule-application routing
   exit
ruledef rr_wsp_co_src_port
   udp src-port = 9201
   rule-application routing
   exit
ruledef dns_tcp_port
   tcp either-port = 53
   rule-application routing
   exit
ruledef dns_udp_port
   udp either-port = 53
   rule-application routing
   exit
ruledef ftp-control-port
   tcp either-port = 21
   rule-application routing
   exit
```

```
ruledef ftp-data-port
   tcp either-port = 20
   rule-application routing
   exit
ruledef pop3-port
   tcp either-port = 110
   rule-application routing
   exit
ruledef smtp-port
   tcp either-port = 25
   rule-application routing
   exit
ruledef rtsp-port
   tcp either-port = 554
   rule-application routing
   exit
ruledef sdp_route
   sip content type = application/sdp
   rule-application routing
   exit
ruledef secure-http-port
   tcp either-port = 443
   rule-application routing
   exit
ruledef sip-port
   udp either-port = 5060
   rule-application routing
   exit
ruledef imap-port
```

```
   tcp either-port = 553
   rule-application routing
   exit
ruledef http-pkts
   http any-match = TRUE
   exit
ruledef http-url-google
   http url contains google
   exit
ruledef http-url-rediff
   http url starts-with http://www.rediff.com/
   exit
ruledef <ip_address>:*/*
   www url starts-with http://<ip_address>/
   www url starts-with http://<ip_address>:
   multi-line-or all-lines
   exit
ruledef mms-pkts
   mms any-match = TRUE
   exit
ruledef rtsp-pkts
   rtsp any-match = TRUE
   exit
ruledef rtp-pkts
   rtp any-match = TRUE
   exit
ruledef pop3-pkts
   pop3 any-match = TRUE
   exit
```

```
ruledef smtp-pkts
   smtp any-match = TRUE
   exit
ruledef ftp-pkts
   ftp any-match = TRUE
   exit
ruledef imap-pkts
   imap any-match = TRUE
   exit
ruledef sip-pkts
   sip any-match = TRUE
   exit
ruledef sdp-pkts
   sdp any-match = TRUE
   exit
ruledef wsp-pkts
   wsp any-match = TRUE
   exit
ruledef wtp-pkts
   wtp any-match = TRUE
   exit
ruledef dns-pkts
   dns any-match = TRUE
   exit
ruledef https-pkts
   secure-http any-match = TRUE
   exit
ruledef udp-pkts
   udp any-match = TRUE
```

```
            exit
         ruledef tcp-pkts
            tcp any-match = TRUE
            exit
         ruledef ip-pkts
            ip any-match = TRUE
            exit
         charging-action standard
            content-id 80
            retransmissions-counted
            billing-action egcdr
            cca charging credit
            exit
         charging-action nocharge
            content-id 81
            retransmissions-counted
            exit
         rulebase consumer-standard
            billing-records egcdr
            action priority 100 ruledef http-url-google charging-action
standard
            action priority 101 ruledef http-url-rediff
 charging-action standard
            action priority 102 ruledef <ip_address>:*/* charging-action
standard
            action priority 1000 ruledef http-pkts charging-action standard
            action priority 1100 ruledef mms-pkts charging-action standard
            action priority 1200 ruledef rtsp-pkts charging-action standard
            action priority 1300 ruledef rtp-pkts charging-action standard
            action priority 1400 ruledef smtp-pkts charging-action standard
```

```
                action priority 1500 ruledef pop3-pkts charging-action standard
                action priority 1600 ruledef dns-pkts charging-action standard
                action priority 1700 ruledef ftp-pkts charging-action standard
                action priority 1800 ruledef sip-pkts charging-action standard
                action priority 1900 ruledef sdp-pkts charging-action standard
                action priority 2000 ruledef imap-pkts charging-action standard
                action priority 2100 ruledef wsp-pkts charging-action standard
                action priority 2200 ruledef wtp-pkts charging-action standard
                action priority 2300 ruledef https-pkts charging-action standard
                action priority 2400 ruledef tcp-pkts charging-action standard
                action priority 2500 ruledef udp-pkts charging-action standard
                action priority 2600 ruledef ip-pkts charging-action nocharge
                route priority 1 ruledef rr_wsp_co_src_port analyzer wsp-
connection-oriented
                route priority 2 ruledef rr_wsp_co_dst_port analyzer wsp-
connection-oriented
                route priority 3 ruledef rr_wsp_cl_src_port analyzer wsp-
connection-less
                route priority 4 ruledef rr_wsp_cl_dst_port analyzer wsp-
connection-less
                route priority 5 ruledef rr_http_80 analyzer http
                route priority 6 ruledef rr_mms_http_ct analyzer mms
                route priority 7 ruledef rr_mms_http_url analyzer mms
                route priority 8 ruledef rr_mms_wsp_ct analyzer mms
                route priority 9 ruledef rr_mms_wsp_url analyzer mms
                route priority 10 ruledef dns_tcp_port analyzer dns
                route priority 11 ruledef dns_tcp_port analyzer dns
                route priority 15 ruledef ftp-control-port analyzer ftp-control
                route priority 16 ruledef ftp-data-port analyzer ftp-data
                route priority 17 ruledef pop3-port analyzer pop3
```

```
            route priority 18 ruledef smtp-port analyzer smtp
            route priority 19 ruledef rtsp-port analyzer rtsp
            route priority 20 ruledef sdp_route analyzer sdp
            route priority 21 ruledef secure-http-port analyzer secure-http
            route priority 22 ruledef sip-port analyzer sip
            route priority 23 ruledef imap-port analyzer imap
            rtp dynamic-flow-detection
            exit
         credit-control
            diameter origin endpoint acs-dcca.starentnetworks.com
            diameter peer-select peer minid.starentnetworks.com
            diameter dictionary dcca-custom1
            pending-traffic-treatment noquota pass
            failure-handling initial-request continue go-offline-after-tx-
expiry
            failure-handling update-request continue
            failure-handling terminate-request continue
            exit
         exit
```