



Overview

The Cisco Content Services Gateway (CSG) is a high-speed processing module that brings content billing and user awareness to the Cisco Catalyst 6500 series switch or Cisco 7600 series router platforms. The CSG is typically located at the edge of a network in an Internet service provider (ISP) point of presence (POP), or Regional Data Center.

In addition to performing standard IP flow accounting, the CSG also examines various protocol requests (wireless application protocol [WAP], e-mail, FTP, Real Time Streaming Protocol [RTSP], and HTTP) to gather URLs and other header information for accounting purposes. Additionally, the CSG gathers information on usernames and usage statistics, and enables differentiated billing for individual transactions based on hostname, on the directory accessed, or on individual files.

The CSG inspects IP traffic at levels deeper than typical routers. When doing so, the CSG behaves partly as a proxy server. Therefore, design your network security strategy to protect the CSG as you would any proxy or server.

This section includes the following information:

- [What's New, page 1-1](#)
- [Features from Previous Releases, page 1-6](#)
- [Prerequisites, page 1-55](#)
- [Restrictions, page 1-55](#)

What's New

The CSG 3.1(3)C7(1) includes the following new features:

- [RADIUS VSA Subattribute Parsing, page 1-2](#)
- [User Table Entry Idle Timer, page 1-2](#)
- [RTSP PAUSE Support, page 1-3](#)
- [SMTP CDR Header Removal, page 1-3](#)
- [Service-Level CDR Support for FTP, RTSP, and WAP 1.x, page 1-4](#)
- [Performance Improvements for WAP, page 1-4](#)
- [IP Fragment Support for IMAP, POP3, and SMTP, page 1-4](#)
- [Support for Out-of-Order Packets for HTTP, IMAP, POP3, and SMTP, page 1-4](#)
- [Support for Additional Services, Services Rules, and Content/Policy Pairs, page 1-4](#)

- [Support for Multipart HTTP](#), page 1-5
- [Support for WAP Segmentation and Reassembly \(SAR\)](#), page 1-5
- [Support for Obscuring the IP Address in X-Forwarded-For Headers](#), page 1-5
- [Enhanced Quota Reconciliation](#), page 1-5
- [Enhancements for Layer 2](#), page 1-6

Additional features are described in the “[Features from Previous Releases](#)” section on page 1-6.

RADIUS VSA Subattribute Parsing

You can specify vendor-specific attribute (VSA) subattributes for RADIUS attribute reporting, and for inclusion in Packet of Disconnect (PoD) messages.

The impact of RADIUS VSA subattribute parsing on CSG performance has not been measured. Storage is consumed based on the attributes selected.

To specify the RADIUS attributes and VSA subattributes to be copied from the RADIUS Start message and sent to the quota server and Billing Mediation Agent (BMA), use the **report radius attribute** command in CSG accounting configuration mode.

To specify the RADIUS VSA subattributes to be copied from the RADIUS Start message and sent to the Network Access Server (NAS) in the PoD message, use the **radius pod attribute** command in CSG user group configuration mode. If RADIUS attribute 26 is included in the message, only the configured subattributes are saved, not the entire attribute. Therefore, only the subattributes specified using the **radius pod attribute** command are reported or included in the PoD messages.

User Table Entry Idle Timer

You can set an idle timer for CSG User Table entries for each individual billing plan, or you can set a global User Table entry idle timer. If an idle timer is set for a billing plan, the CSG uses that idle timer. Otherwise, the CSG uses the global idle timer. That is, if there is an entry idle timer value in the billing plan, it is used; otherwise, if there is a global entry idle timer value configured, it is used.

The idle timer for a user entry starts when all billable sessions for that user have ended. Typically, a TCP session ends when the client and the server have sent FIN messages to each other. Other protocols time out based on the configured idle timer value in the content configuration. The timer restarts when a RADIUS Accounting Start or an Interim Accounting message is received. The timer stops when a session starts.

The idle timer also enables the CSG to eliminate an idle User Table entry if the NAS fails to deliver a RADIUS Accounting Stop request for an idle user. Eliminating idle entries from the User Table frees up CSG resources.

When the idle timer expires, if Packet of Disconnect (PoD) is not configured, the CSG deletes the User Table entry. If Packet of Disconnect (PoD) is configured, the CSG sends a PoD message and the CSG deletes the User Table entry when the PoD message is ACKed, NAKed, or when all retries have been sent; the RADIUS Stop message does not have to be received by the CSG.

If Connection Duration Billing is enabled, you can use either the billing plan entry idle timer or the user group entry idle timer to release a user connection.

To set a global User Table idle timer, use the **entries idle** command in CSG user group configuration mode.

To set a User Table idle timer for a specific billing plan, use the **entries idle** command in CSG billing configuration mode.

RTSP PAUSE Support

The CSG monitors the RTSP control session between the RTSP client and server and scans for PAUSE and PLAY methods. When a PAUSE method is detected, the CSG initiates an event to inform the prepaid service to stop charging for duration-based billing. Then, when a PLAY method is detected, the CSG initiates an event to inform the prepaid service to resume the charging for duration-based billing. The event corresponding to the PLAY is only necessary when the CSG is in the PAUSE state.

When configuring RTSP PAUSE support, keep the following considerations in mind:

- Duration-based billing applies only to a billing service, and RTSP might not be the only application that is operating over a given billing service. Therefore, the suspension of billing for the PAUSE period applies only if there are no other applications operating over the same billing service.
- Both last billable and intermediate idles for RTSP are excluded from duration-based billing if RTSP PAUSE support is configured.
- RTSP PAUSE support applies only to classical RTSP transport, in which the stream data is transmitted over a separate User Datagram Protocol (UDP) connection. RTSP PAUSE support does not apply to the TCP or HTTP interleaved transport modes.
- RTSP pause support is independent of the UDP content idle timer value and of the RTSP service idle timer. RTSP pause support does not stop the UDP content idle timer.
- RTSP PAUSE support applies only to charges for prepaid quota usage (quadrans TLV).
- The RTSP PAUSE time is not reflected in any CDRs.

To exclude the RTSP PAUSE time from the duration-based billing calculation, use the **meter exclude rtsp pause** command in CSG service configuration mode.

SMTP CDR Header Removal

An SMTP CDR can be very large, as it includes a report attribute for each SMTP header embedded at the beginning of a message. The CSG enables you to eliminate these headers from an SMTP CDR, leaving only the SMTP envelope headers and the size attribute in the report. These are reported as X-CSG-MAIL, X-CSG-RCPT and X-CSG-SIZE.

To enable SMTP CDR header removal, set the `CSG_SMTP_CDR_HEADER_REDUCTION` variable to 1 using the **variable** command in module CSG configuration mode. When SMTP CDR header removal is enabled, the CSG reports the following header information to the BMA:

- X-CSG-MAIL
- X-CSG-RCPT
- X-CSG-SIZE

To restore the SMTP CDR headers, set the `CSG_SMTP_CDR_HEADER_REDUCTION` variable to 0. When SMTP CDR headers are restored, the CSG reports the following header information to the BMA:

- X-CSG-MAIL
- X-CSG-RCPT
- X-CSG-SIZE

- X-Priority¹
- X-MSMail-Priority¹
- X-Mailer¹
- X-MimeOLE¹

Service-Level CDR Support for FTP, RTSP, and WAP 1.x

The CSG now provides service-level CDR support for FTP, RTSP, and WAP 1.x.

There are no commands required to enable service-level CDR support for FTP, RTSP, and WAP 1.x.

Performance Improvements for WAP

Internal processing improvements have resulted in increased WAP performance.

There are no commands required to enable these performance improvements.

IP Fragment Support for IMAP, POP3, and SMTP

The CSG now supports IP fragmentation for IMAP, POP3, and SMTP, including fragmentation that arrive out of order. This is in addition to the existing IP fragmentation support for HTTP, WAP 2.0, WAP 1.x, and generic Layer 4 flows.

The CSG does not support IP fragmentation for FTP, for RTSP control connections, nor for RADIUS flows. The CSG drops IP fragments for those unsupported protocols.

There are no commands required to enable IP fragment support for IMAP, POP3, and SMTP.

Support for Out-of-Order Packets for HTTP, IMAP, POP3, and SMTP

The CSG buffers packets that are received out of order and processes them in the proper order. This does not guarantee that the CSG transmits the packets in any specific order, only that the CSG can handle them arriving out of order.

There are no commands required to enable support for out-of-order packets.

Support for Additional Services, Services Rules, and Content/Policy Pairs

The CSG supports 1024 configurable services, up from 255.

The CSG supports 4096 services rules, up from 1024.

The CSG supports up to 4096 content/policy pairs configured under services within a billing plan, up from 1024 pairs.

There are no commands required to enable these increases.

1. Presence of this header depends on the contents of the SMTP header.

Support for Multipart HTTP

For HTTP sessions, multipart content does not cause the CSG to invoke Layer 4 billing for the remainder of the connection. Instead, the CSG parses the data for the delimiter specified in the header and continues to use Layer 7 billing.

There are no commands required to enable support for multipart HTTP.

Support for WAP Segmentation and Reassembly (SAR)

Prior to the CSG 3.1(3)C7(1), if the first packet of a WAP transaction did not contain the full URL, the CSG allowed the entire transaction to pass through at no charge. In the CSG 3.1(3)C7(1), the CSG applies the appropriate policy to the WAP transaction if it contains a URL that spans multiple WAP segmented packets.

There are no commands required to enable support for WAP SAR.

Support for Obscuring the IP Address in X-Forwarded-For Headers

To prevent exposure of potentially sensitive IP addresses, the CSG can obscure the contents of X-Forwarded-For headers, overwriting the contents with blanks.

- If you want to obscure the contents of the X-Forwarded-For header, use the **variable** command in module CSG configuration mode to set the `CSG_OBSCURE_X_FORWARDED_FOR` environment variable to 1.
- If you do not want to obscure the contents of the X-Forwarded-For header, set the `CSG_OBSCURE_X_FORWARDED_FOR` to 0 (the default setting).

If your configuration is fault-tolerant, keep the following considerations in mind:

- Use the **variable** command in module CSG configuration mode to set the `CSG_FT_CONTENT` environment variable to 1. That is, replicate sessions only if replication is configured in the content.
- Do not configure the **replicate connection tcp** command in CSG content configuration mode.

Enhanced Quota Reconciliation

During internal quota reconciliation, the CSG might drop packets for some prepaid users, which can affect user throughput.

To prevent this problem, set the `CSG_QUOTA_BLOCK` environment variable to 0, using the **variable** command in module CSG configuration mode. Setting this variable to 0 enables the CSG to forward packets for a prepaid user during quota reconciliation, regardless of whether quota is available for the user. When quota reconciliation is complete, if the CSG determines that the user has no quota available, the CSG terminates the connection.

The CSG supports enhanced quota reconciliation for all accounting types.

If you want the CSG to continue to drop packets that arrive during quota reconciliation, set the `CSG_QUOTA_BLOCK` to 1. This is the default setting.

Enhancements for Layer 2

The CSG processes “Hello” messages for HSRP version 1 and version 2.

If the MAC address of a directly-attached client is unknown, the CSG uses the Address Resolution Protocol (ARP) to get the IP address of the client dynamically. A “dummy” route for the client is no longer needed.

There are no commands required to enable these enhancements for Layer 2.

Features from Previous Releases

In addition to new features introduced in this release, the CSG provides the following features and functionality that were introduced prior to the CSG 3.1(3)C7(1):

- [Byte Counting, page 1-7](#)
- [CDR Support, page 1-10](#)
- [Quota Support, page 1-13](#)
- [Advice of Charge and Related Features, page 1-16](#)
- [Service Verification, page 1-20](#)
- [CSG User Table, page 1-20](#)
- [RADIUS Features, page 1-21](#)
- [HTTP Features, page 1-26](#)
- [WAP Features, page 1-28](#)
- [RTSP Features, page 1-30](#)
- [IMAP Support, page 1-34](#)
- [POP3 Support, page 1-37](#)
- [SMTP and POP3 Billing, page 1-38](#)
- [FTP Billing, page 1-39](#)
- [Header Mapping and URL Mapping, page 1-39](#)
- [Passthrough Mode and the Default Quota, page 1-39](#)
- [Service Duration Billing, page 1-41](#)
- [Connection Duration Billing, page 1-45](#)
- [Postpaid Service Tagging, page 1-46](#)
- [Stateful Redundancy and Failover, page 1-46](#)
- [“Default” Policy, page 1-47](#)
- [Prepaid Error Reimbursement, page 1-48](#)
- [Support for the Cisco Persistent Storage Device, page 1-48](#)
- [Postpaid Billing, page 1-49](#)
- [BMA Load Sharing, page 1-49](#)
- [Prepaid Content Billing and Accounting, page 1-50](#)
- [Obtaining User IDs, page 1-51](#)

- [Filtering Accounting, page 1-51](#)
- [Intermediate Billing Records, page 1-52](#)
- [Packet Forwarding, page 1-52](#)
- [Supplemental Usage Reports, page 1-53](#)
- [Enhanced Interoperability with Cisco Service-Aware GGSN, page 1-53](#)
- [Miscellaneous Features, page 1-53](#)

Byte Counting

The CSG reports the number of IP bytes uploaded and downloaded, the number of TCP bytes uploaded and downloaded by the application, and the packet counts (or PDU counts for WAP records). These counts exclude the IP and TCP headers, as well as retransmissions.

This section includes the following information:

- [HTTP Byte Counting, page 1-7](#)
- [WAP Byte and Packet Counting, page 1-8](#)
- [IMAP Byte Counting, page 1-9](#)
- [FTP and RTSP Byte Counting, page 1-10](#)
- [POP3 and SMTP Byte Counting, page 1-10](#)
- [Byte and Packet Counting After a Failover, page 1-10](#)

HTTP Byte Counting

HTTP 1.1 allows a client to send multiple HTTP requests without waiting for the corresponding responses. Therefore, a single IP datagram might contain requests or responses for more than one HTTP transaction.

The CSG counts the IP bytes for the TCP session SYN as part of the first transaction. The CSG counts the IP bytes for the TCP session FIN, FIN/ACK, or RST as part of the last transaction. The CSG counts the IP and TCP header bytes of an IP datagram that contains multiple transactions as part of the first transaction in the datagram.

If the CSG receives retransmitted SYN packet before it receives the first SYN/ACK from the server, it does not include the IP bytes for the retransmitted SYN packet in the byte counts in the HTTP_Stats CDR. This is a timing condition.

The CSG discards out-of-order FIN packets, even if they contain data, and depends on retransmission of the out-of-order FIN packets to ensure correct billing.

If the final ACK on a TCP 3-way handshake is retransmitted, the CSG does not report the IP bytes associated with the retransmitted ACK in the HTTP_Stats CDR.

To enable the CSG to count fixed-format HTTP IP bytes more accurately, a new CDR and a new TLV have been added to the existing fixed HTTP intermediate CDRs.

There are no commands required to enable HTTP IP byte counting.

This section includes the following additional information:

- [HTTP IP Bytes vs. TCP Bytes, page 1-8](#)

- [Counting Uncorrelated HTTP IP Bytes, page 1-8](#)
- [Packet Counts for Pipelined HTTP, page 1-8](#)

HTTP IP Bytes vs. TCP Bytes

In previous releases, the CSG reported IP bytes the same as TCP bytes for an HTTP transaction. Beginning in release 3.1(3)C7(1), the CSG reports the total number of IP bytes of an HTTP transaction transferred between a client and a server. As a result of these changes, properly billed HTTP Layer 7 transactions might be more expensive than in previous CSG releases. For a given transaction, there will always be more IP bytes than TCP bytes.

For HTTP quota management, the billing process has always managed IP bytes, not TCP bytes (for **basis byte ip**) and has always provided transaction grants as IP bytes.

From a system behavior point of view, the CSG quota management for HTTP may appear different, because the forwarding process takes the granted quota and applies it to IP bytes instead of TCP bytes.

For example, an empty TCP packet for HTTP would have previously consumed 0 bytes of IP quota - now it would take 40 (assuming standard IP and TCP header size).



Note

If you want the CSG to continue to report TCP byte counts for HTTP transactions, you can configure a service with **basis byte tcp** to count TCP bytes instead of IP bytes as quadrants, and you can configure the CSG to inspect BMA records for reported TCP byte counts.

Configuring **basis byte tcp** allows counting of only TCP payload and exclusion of overhead for network retransmission. With this option, the CSG excludes IP and TCP headers from volume counts. Retransmitted packets are also not counted.

Counting Uncorrelated HTTP IP Bytes

Sometimes the CSG cannot correlate some IP bytes to any transaction at the end of a TCP session. This can include any retransmits or ACKs without payloads that are received after the CSG has reported the CDR for a specific transaction. You can configure the CSG to include these IP bytes in its reports by setting the `CSG_HTTP_STATS_DELAY` environment variable to a non-zero value. To do so, use the **variable** command in module CSG configuration mode.

Packet Counts for Pipelined HTTP

Packet counts for pipelined HTTP operations are a snapshot of the number of packets detected on the connection since the previous statistics were reported. The packet count might even be zero if two pipelined operations share the same packet.

WAP Byte and Packet Counting

WAP byte counting is always IP-based.

The CSG reports WAP datagram sizes (including IP and UDP headers), the number of IP packets per transaction, and PDU counts. (The PDU count is not the same as the packet count. Multiple WAP PDUs can share a single packet.)

Bytes for retransmitted WAP PDUs and segments are not counted against quota, but they are counted and listed separately from non-retransmitted counts in the WAP CDRs.

Byte and PDU counts are further specified by source. Reports include the number of bytes and PDUs uploaded from source to destination and the number of bytes downloaded from destination to source.

The CSG splits all concatenated PDUs received from the client into multiple IP packets to be sent to the server. Therefore, packet counts are based on the number of WAP PDUs, not on the number of IP packets.

Byte counting for concatenated PDUs is complicated because multiple transactions are combined into a single IP packet. For example, a concatenated CONNECT/GET shares the same IP/UDP headers, yet they are treated as two separate transactions, they result in two separate CDRs, and they might even be charged differently from each other. In addition to the IP/UDP headers, there are several other bytes in the packet that define it as a concatenated packet. It might not be obvious to which transaction these bytes are assigned. Here is how the CSG assigns the IP bytes:

- The size of the IP/UDP headers (usually 28 bytes) is assigned to the first PDU.
- The single byte that identifies the packet as a concatenated packet is also be assigned to the first PDU.
- A one- or two-byte length field is assigned to each PDU.

For example, a CONNECT/GET concatenated PDU that contains one-byte PDU length fields yields the following byte count totals:

- CONNECT transaction = IP/UDP header length + 1 + 1 + PDU size
- GET transaction = 1 + PDU size

IMAP Byte Counting

Service-level fixed format CDRs for IMAP include the following IMAP-specific counts:

- Number of header retrievals. That is, the number of times that the CSG retrieved the header attribute of an e-mail message (for example, **BODY[HEADER]**, **RFC822.HEADER**).
- Header IP bytes sent upstream (client to server)
- Header IP bytes sent downstream (server to client)
- Header TCP bytes sent upstream
- Header TCP bytes sent downstream
- Number of body retrievals. That is, the number of times that the CSG retrieved any portion of the body text of an e-mail message (for example, **BODY[]**, **BODY[TEXT]**, **BODY[3]**, **BODY[<0.4096>**, **RFC822**, **RFC822.TEXT**).
- Body IP bytes sent upstream
- Body IP bytes sent downstream
- Body TCP bytes sent upstream
- Body TCP bytes sent downstream

The CSG reports incremental byte counts for the IMAP service-level fixed format CDRs. For example, if 100 KB of traffic is generated for the first 15 minutes, 50 KB for the next 15 minutes, and the CSG generates intermediate CDRs every 15 minutes, then the CSG reports the change in the total byte count from the point at which the last CDR was reported to the point at which the current CDR is reported. Thus, the first CDR would report 100 KB, and the second would report 50 KB.

With fixed format CDRs, the incremental byte counts might be reported at a given time interval or after a volume threshold has been reached (for example, every 15 minutes, or after every 100 KB.)

For IMAP byte counting, keep the following considerations in mind:

- Message tags cannot be longer than 100 bytes. If the CSG encounters a message with a tag that is longer than 100 bytes, only the IP and TCP upstream and downstream byte counts are reported.
- The byte counts associated with a continuation response flow are accounted for in the next classified transaction.

FTP and RTSP Byte Counting

For FTP and RTSP, the CSG reports the upstream and downstream IP bytes and TCP bytes.

Even though FTP and RTSP data sessions usually appear to be network-initiated, the uploaded bytes for FTP and RTSP (for example, in IP statistics) are counted from the originator of the session, the endpoint from which the first packet for the session is received.



Note

For service-level CDRs, the uploaded bytes for FTP and RTSP are counted from the subscriber to the network, and the downloaded bytes are counted from the network to the subscriber.

The CSG discards out-of-order FIN packets, even if they contain data, and depends on retransmission of the out-of-order FIN packets to ensure correct billing.

POP3 and SMTP Byte Counting

For POP3 and SMTP, the CSG reports the upstream and downstream IP bytes and TCP bytes.

Byte and Packet Counting After a Failover

After a failover, if the first packet received by the standby CSG (now the active CSG) is a retransmitted packet, the TCP byte count might be higher than it should be. However, the IP byte count is calculated correctly and therefore might be less than the TCP byte count. If this occurs, the CSG reports the TCP byte count as 0 in the final CDR.

CDR Support

The CSG provides the following call detail record (CDR) support:

- [Fixed CDR Support for HTTP, page 1-11](#)
- [Fixed CDR Support for IMAP, page 1-11](#)
- [Fixed CDR Support for RTSP, page 1-12](#)
- [Single CDR Support for WAP Connectionless and HTTP Traffic, page 1-12](#)
- [Service-Level CDR Summarization, page 1-12](#)
- [Prepaid and Postpaid Envelope Information Support for SMTP, page 1-13](#)
- [Fixed Attribute CDRs for WAP, page 1-13](#)

Fixed CDR Support for HTTP

The CSG provides fixed call detail record (CDR) support for HTTP as well as for WAP. This support generates one fixed CDR for every HTTP transaction, instead of two CDRs, which are typically generated at the beginning and end of the transaction.

The single CDR contains all the fields that are included in the HTTP header and HTTP statistics records, in a fixed format. In addition, the same fixed-format service TLVs that were included for WAP are also included for HTTP.

The single CDR also includes RADIUS TLVs, in ascending order, based on the RADIUS TLVs configured using the **report radius attribute** command in CSG accounting configuration mode. This is a change from the CSG 3.1(3)C5(1), in which you configured up to 10 specific RADIUS attributes to be included in the CDR in a predefined order. This scheme is very flexible, enabling you to add RADIUS attributes dynamically. This change in the handling of RADIUS TLVs applies to both WAP and HTTP fixed CDRs.

Fixed CDR support does not support RADIUS attribute 26 (the vendor-specific attribute, or VSA), because the list of attributes defined within the VSA is variable. Therefore, a predefined “fixed” list of attributes cannot be determined when RADIUS attribute 26 is configured.

To enable the fixed format feature for HTTP and for WAP, use the **records format fixed** command in CSG accounting configuration mode.

The CSG also supports fixed HTTP intermediate records. The fixed intermediate record format is identical to the format of the fixed record created at the end of the transaction, except for the message type, which is necessary to differentiate the two records. The intermediate statistics, such as TCP byte counts, are per intermediate period, and are not cumulative. This differs from the existing HTTP intermediate support for variable format CDRs, in which the TCP byte counts are cumulative.

The Content Delivered TLV contains a value of 0x00 (not delivered) if the HTTP response code is greater than or equal to 400, or if the TCP byte download count is less than 12 bytes.

Fixed CDR Support for IMAP

The CSG now supports postpaid billing for the Internet Message Access Protocol (IMAP), in addition to postpaid billing for Post Office Protocol, version 3 (POP3), and Simple Mail Transfer Protocol (SMTP). This fixed CDR support for IMAP enables the CSG to report service-level fixed format CDRs for IMAP.

To enable the CSG to support IMAP, use the **records format fixed** command in CSG accounting configuration mode, and the **accounting type imap** command in CSG policy configuration mode.

When configuring CSG support for IMAP, keep in mind the following considerations:

- The CSG supports only postpaid billing for IMAP. IMAP transactions for a prepaid user are treated as postpaid.
- The CSG does not support AoC for IMAP. If AoC is configured for an IMAP user, AoC is ignored for that user.
- The CSG cannot examine IMAP flows sent over an encrypted tunnel, such as Secure Socket Layer (SSL) or Transport Layer Security (TLS). When an encrypted tunnel is used for IMAP traffic, the CSG records only IP and TCP upstream and downstream byte counts. No other counts are provided.

Fixed CDR Support for RTSP

This feature enables the CSG to send the existing RTSP stream CDRs in a fixed format. The same fixed format service TLVs that were included for WAP are also included for RTSP.

To enable the fixed format feature for RTSP, use the **records format fixed** command in CSG accounting configuration mode.

Single CDR Support for WAP Connectionless and HTTP Traffic

The CSG already reduces the multiple CDRs generated for WAP connection-oriented traffic to a single CDR, which is reported at the end of the transaction. This feature is extended to WAP connectionless traffic and HTTP traffic.

The single CDR contains the standard variable format, and it also includes a comprehensive list of TLVs containing all pertinent information for the transaction. For WAP connectionless transactions, it includes everything that is included in a WAP GET and REPLY CDR. For HTTP transactions, it includes everything in the HTTP header and HTTP statistics records.

To enable single CDR support for WAP connection-oriented, WAP connectionless, and HTTP traffic, use the **variable-single-cdr** keyword with the **records format** command in CSG accounting configuration mode.

When you configure single CDR support, the CSG suppresses HTTP intermediate record generation.

Service-Level CDR Summarization

By default, the CSG generates billing records for each transaction. This has the potential to overwhelm the charging gateway (CG) or the collector. To prevent this situation, the CSG can summarize CDRs at the service level, instead of at the transaction level.

For example, if a user accesses the open Internet service, and the data is billed solely on the basis of volume, it is of little use to generate records for each HTTP transaction. With service-level CDR summarization enabled, the CSG generates only consolidated records on service-level usage. Information from individual events is not reported (for example, no URLs).

The CSG uses the Service Usage - variable format (0x0040) CDR for service-level CDR summarization.

Service-level CDRs differ from Service Stop CDRs as follows:

- The CSG sends a Service Stop CDR to the quota server when a prepaid service instance ends. At the same time, the CSG sends a companion CDR, the Service Stop Notification CDR (0x11), to the BMA.
- The CSG sends a service-level CDR to the BMA when a service instance ends, or, if configured, when a volume- or time-based threshold is met. The CSG sends service-level CDRs only to the BMA, and only if the service is configured for service-level CDRs.

The CSG can generate intermediate service-level CDRs for volume-based billing or for time-based billing. Cause codes 0x0A and 0x0B are used for service-level CDRs generated for volume- or time-based billing.

The CSG supports the following protocols in both fixed and variable format: IP, HTTP, SMTP, POP3, and IMAP. (POP3 and IMAP are supported in postpaid mode only.)

Service-level CDRs are generated only for subscribers with entries in the CSG User Table entry. If a subscriber does not have an entry in the User Table, the CSG generates transaction-level CDRs.

To enable service-level CDR summarization, use the **records granularity** command in CSG service configuration mode.

To enable service-level CDR summarization in postpaid mode, you must specify that the associated billing plan is postpaid by using the **mode postpaid** command in CSG billing configuration mode.

Prepaid and Postpaid Envelope Information Support for SMTP

The CSG provides SMTP with prepaid and postpaid support, including envelope information in the CDR.

SMTP prepaid support includes all existing billing options (including IP bytes, TCP bytes excluding retransmissions, duration, and fixed). SMTP CDRs include e-mail envelope information as well as IP byte counts, TCP byte counts, and e-mail data (X-CSG-SIZE) byte counts for each e-mail message. When multiple e-mails are sent over a single TCP connection, each e-mail message is assigned byte counts until the start of the next e-mail message. The last e-mail is assigned bytes from the start of that e-mail until the end of the TCP connection.

The return code reported in the CDR is the one returned for the DATA portion of the e-mail message. If the CSG does not receive that data return code, it reports the last error return code (other than **250**) received for individual recipients (because a bad recipient return code might be the cause of the e-mail not being sent). If the CSG receives a QUIT before receiving any return code, it reports a default return code of **554** (Transaction failed). This enables the CSG to apply refunding via the SMTP return code value.

If the user runs out of quota in the middle of a transaction, the session is terminated and all known information is reported in a CDR. The application return code indicates whether the e-mail was received, and the authentication failure bit is set in the TCP **flags** field.

The CSG no longer uses the TCP Stats CDR, which was generated at the end of the TCP connection, because the information in the TCP Stats CDR duplicates the information in the SMTP CDR.

Fixed Attribute CDRs for WAP

To support some legacy billing systems, the CSG provides a fixed attribute format for WAP CDRs. The same set of attributes is reported in each CDR regardless of the Wireless Session Protocol (WSP) protocol data unit (PDU) type. CDRs contain zero-length attributes when there is no information to report, but the same set of attributes are always reported in the same sequence.

Quota Support

The CSG provides the following quota support features:

- [Quota Server Load Sharing, page 1-14](#)
- [Quota Push, page 1-14](#)
- [Quota Balance Replacement, page 1-14](#)
- [Delayed Quota Reauthorization, page 1-15](#)
- [Configurable Reauthorization Threshold, page 1-15](#)
- [Tariff Switch, page 1-15](#)
- [Asynchronous Quota Return, page 1-16](#)

Quota Server Load Sharing

The CSG supports multiple quota servers. This support is useful in environments in which the number of quota transactions sent by the CSG could overwhelm a single quota server. Multiple quota servers can be simultaneously active, and the CSG assigns a quota server to each user. All quota transactions for the user are handled by the same quota server.

When a quota server fails, all the users associated with that quota server are distributed among other quota servers.

**Note**

Multiple quota servers cannot have the same IP address.

Quota Push

This feature enables operators to “push” quota for a service instance associated with an individual user. This enables quota servers to provide quota for a user or service before traffic from that user or service reaches the CSG. This eliminates the delay that can occur when quota is obtained through a service authorization request and response. A sophisticated quota server could also use quota push for better control of quota levels during active sessions.

The CSG accepts a quota push for a user at any point after the user and billing plan are known to the CSG (that is, when a User Table element exists for the user). For example, the CSG accepts a quota push after receiving an accounting start but does not require an existing service for the user (one is created). The CSG does not begin charging against quota until traffic begins to arrive and a session is created. Zero-quota might be granted so that cause code and authorization actions can be set (for example, for a free service). A quota download message is sent to the BMA in response to receiving a quota push.

The service idle timer starts whenever quota is pushed, in case the expected traffic never arrives.

There are no commands required to enable quota push.

Quota Balance Replacement

By default, when the CSG receives a quota grant from the quota server, the CSG adds the granted quota to the current balance for a subscriber’s service. Quota balance replacement enables the quota server to instruct the CSG to replace the current quota balance with the amount of granted quota for a subscriber’s service. Note that if the replacement grant is provided in a Service Authorization Response, the CSG subtracts the amount of quota used since the Service Reauthorization Requests from the granted quota before replacing the balance.

There are no commands required to enable quota balance replacement.

Delayed Quota Reauthorization

The CSG accepts the Reauthorization Delay TLV, which specifies the number of seconds the CSG delays its next reauthorization request to the quota server for the service specified in the message. This TLV also specifies the action the CSG is to take for the service between the time the message is received and the next reauthorization:

- **Wait**—The CSG keeps transactions in a pending state during the delay period. In pending state, the CSG maintains the transaction state but drops packets.
- **Deny**—The CSG drops new transactions during the delay period. Existing transactions are dropped if quota expires during the delay period. The CSG does not maintain the session state; the user must open a new connection after the delay period.



Note For HTTP pipelining, dropping new transactions can also affect existing transactions if they share the same TCP connection.

Quota servers can use delayed quota reauthorization to deny subscribers access to CSG categories without having to continually deny authorization requests (that is, for blacklisting services). To do so, the quota server sends a grant of 0 quadrans in a Service Authorization Response, Quota Push Request, or Service Verification Response message, with a long reauthorization delay timer (0xFFFFFFFF), a Deny action, and a cause code of 0x03.

There are no commands required to enable delayed quota reauthorization.

Configurable Reauthorization Threshold

You can configure the thresholds of available quota that trigger service reauthorization by specifying the following CSG variables on the **variable** command in module CSG configuration mode:

- `CSG_BASIS_BYTE_LOW_QUOTA_MAX`
- `CSG_BASIS_FIXED_LOW_QUOTA_MAX`
- `CSG_BASIS_SEC_LOW_QUOTA`

You can configure the thresholds for volume-, time-, and transaction-based billing, and the CSG can also accept a threshold specified by the quota server in a quota grant to the CSG. The threshold must appear in each and every quota server response.

Tariff Switch

Tariff switch is a feature in which the CSG2 tracks the total quota usage for a prepaid service at the time of a tariff-time change. The tariff-time change is specified on a per-service basis by the quota server.

The CSG2 supports tariff switch for all prepaid protocols. The CSG2 sends tariff switch TLVs only for transaction-level CDRs, not for service-level CDRs.

The tariff switch usage for the prepaid service is reported to the quota server in the next Service Reauthorization, Quota Return, or Service Stop message sent for this service. The tariff switch usage for a tariff-time change is sent once and is sent in addition to the cumulative usage at the time of the message to the quota server. If the Supplemental Usage Reporting feature is enabled for a service at the tariff switch time, the CSG also reports supplemental usage at the tariff switch time in addition to the tariff-time switch usage.

The life of a prepaid service instance might span multiple tariff switch times. For a given service instance associated with a user, the CSG can handle only one tariff switch time value.

The CSG might not generate a Service Reauthorization Request between tariff switch times. The quota server can force a report of the tariff switch time usage by specifying a quota timeout value in the Service Authorization Response that will force a Quota Return before the next tariff switch time. The quota server must choose the timeout carefully to avoid causing a flood of Quota Return messages at a given time. At any given time, the CSG service can track usage for a single tariff switch time; after reporting the usage for a tariff switch, the quota server can specify the time of the next tariff-time change in a Service Authorization Response.

Tariff switch usage for individual transactions is reported to the BMA in the records containing quota usage (typically intermediate and stats records). Note that intermediate records might be sent to the BMA to report tariff switch usage, even without configuration of intermediate records; this is necessary because transactions might span multiple tariff switch times. To avoid flooding the BMA with records, the CSG sends intermediate records to the BMA for transactions that span a tariff switch time and do not terminate quickly.

The “tariff switch qualified usage” field in the CDR is the cumulative usage since the creation of the service instance.

There are no commands required to enable tariff switch. The output of the **show module csg accounting users all detail** command displays information about the tariff switch.

Asynchronous Quota Return

The Asynchronous Quota Return feature allows the quota server to request the CSG to return quota for a defined user and service, and to send a Quota Return.

Prior to CSG 3.1(3)C6(2), the quota returned via a Quota Return message did not include quota reserved for ongoing transactions. Beginning in CSG 3.1(3)C6(2), the quota reserved for ongoing transactions is recalled from the transactions and included in the quota returned in the Quota Return message.

The Quota Return message can include the following fields: “remaining quota”, “qualified quadrans”, “qualified usage”, “qualified quota remaining”, “reserved quota”, and “pending quota”. For fixed-cost billing, the “reserved quota” field is not included. For time-based billing, neither the “reserved quota” field nor the “pending quota” field is included. The “qualified usage” field identifies quota that is known to have been consumed, and is the only value that is safe to commit to a back-end system.

Because all of the quota is returned, there is no longer any reserved quota to process ongoing pending transactions. Packets received for pending transactions are dropped. However, time-based billing holds back 5 seconds of quota, so transactions can proceed while returning quota for time-based billing.

Advice of Charge and Related Features

Advice of Charge (AoC) is a function that enables a service provider to provide messaging and authorization prompts to its subscribers. The CSG’s support for AoC uses a quota server and a customer-provided notification server to host the actual messaging:

- The quota server is responsible for telling the CSG to block client requests and redirect them to the notification server when the client must make a decision to pay for the service. It is also responsible for telling the CSG to allow the client request to flow when the client has agreed to pay.
- The notification server is responsible for communicating fees to the client and providing the option to pay. The client’s payment decision must be communicated from the notification server to the quota server.

The CSG's role in the AoC process is to redirect user data requests to the notification server. The CSG provides two modes for redirecting to the notification server. The choice of mode, and the requirements that mode imposes on the notification server, varies by protocol. The two modes supported by the CSG are:

- URL-redirect—Supported for HTTP and WAP/WSP
- NAT-redirect—Supported for all other protocols

A complete AoC implementation depends heavily on the notification server. With URL-redirect, the notification server can be a standard web server, because the CSG does the redirection at the protocol level. This is the easiest deployment to implement. With NAT-redirect, the CSG just forwards the connection directly to the notification server. The notification server must therefore be able to accept the packets, interpret the protocol, drive an AoC on its own, and properly manage the rest of that user's flow for that connection.

The CSG allows the redirection for AoC to be triggered once per service (when the first access to the service is made by the subscriber), or at the start of any new data transaction. The former is accomplished using the CSG's service verification function, the latter using the CSG's content authorization function. The URL can be pre-configured, or it can be provided dynamically by the quota server (the more flexible option). You can configure content authorization to request a pass/fail authorization for any transaction (for example, for individual SMTP e-mails), but the CSG does not honor redirect requests from the quota server in the middle of a TCP connection.

In general, the method by which the notification server communicates success or failure of the AoC to the quota server is outside the scope of the CSG's role in the process. However, the CSG does provide some additional assists for URL-redirects that greatly ease the burden on the backend systems. For example, the CSG provides the ability to strip trailing tokens from a URL. Therefore, an HTTP-based notification server can be deployed such that it will append the results of the AoC to the user HTTP request when redirecting the subscriber to the final requested content. The CSG reports this URL, token and all, to the quota server on the next Content Authorization Request. If configured to do so, upon successfully receiving permission from the quota server to forward the flow, the CSG strips the token from the request so that the content server is not confused by the extra data.

You can instruct the CSG to obtain authorization from the quota server for each subscriber request for content.

The CSG's support for AoC has the following restrictions:

- The CSG supports AoC via content authorization and URL-redirect for only HTTP and WAP/WSP.
- The CSG does not support AoC for IMAP. If AoC is configured for an IMAP user, AoC is ignored for that user.
- The CSG does not support AoC for Connection Duration services.
- When performing AoC for a TCP connection carrying pipelined HTTP requests, the CSG responds with the redirect to the client as soon as the quota server requests the redirect. This could result in the redirect arriving at the client before responses for previous requests arrive, and the client might associate the redirect with a different request in the pipeline.

To enable the CSG's support for AoC, use the **authorize content** command in CSG service configuration mode.

The CSG provides the following AoC-related features:

- [URL Redirect, page 1-18](#)
- [URL Rewriting, page 1-18](#)
- [WAP URL Appending, page 1-19](#)
- [SMTP Content Authorization, page 1-19](#)

- [Redirect Flexibility, page 1-20](#)
- [Service Verification, page 1-20](#)

URL Redirect

In a redirect scenario, the CSG responds to the HTTP or WAP client with a response code and a URL to which the client must redirect. You can configure the redirect URL using the **redirect** command in CSG user group configuration mode, or the quota server can provide the redirect URL during service authorization (or reauthorization) or content authorization processing.

For service authorization or content authorization, the quota server reply contains the REDIRECT-URL action code and the redirect URL. In some network configurations, you might want the quota server to return a single redirect URL for both WAP and HTTP. If you do not want to use a single redirect URL, the service authorization and Content Authorization Requests identify whether the request is for HTTP or for WAP.

A redirect URL that is returned from the quota server in a service authorization response, or that is returned in a Content Authorization Response with the REDIRECT_URL action code, takes precedence over a redirect URL that is configured using the **redirect** command. The CSG uses the **redirect**-specified URL when the quota server responds with the FORWARD action code.

To control the amount of time and the number of redirects that the CSG allows, specify the following environment variables using the **variable** command in module CSG configuration mode:

- **CSG_REDIRECTS_INTERVAL**—Defines the length of time for which the CSG must redirect.
- **CSG_REDIRECTS_MAX**—Defines the number of requests that are redirected before the CSG stops redirecting, but within the interval time.

The CSG starts the interval timer when the first request is redirected when it receives no quota. The counter is reset, and the timer is stopped after another quota grant of zero is given.

For example, if **CSG_REDIRECTS_MAX** is set to 15 and **CSG_REDIRECTS_INTERVAL** is set to 8 seconds, and the CSG receives a Service Authorization Response with zero quadrans, and the CSG has redirect information, then redirection occurs when the user runs out of quota (assuming that the user has not received quota in the interim). The **CSG_REDIRECTS_INTERVAL** 8-second timer starts after the first redirect. Therefore, request 1 is redirected, and up to 14 more requests can be redirected, if they occur within 8 seconds after the first redirect.

URL Rewriting

When direct communication is not possible between the quota server and the notification server, payment decision information can be shared indirectly by modifying the URL in the client request. The notification server appends a string beginning with a token to the originally requested URL and sends it to the client as part of a redirect reply after the client agrees to pay. The CSG receives the subsequent GET request containing the rewritten URL and sends it to the quota server in a Content Authorization Request. The quota server recognizes the token string and understands that the client has agreed to pay for the request. It responds to the CSG with a FORWARD action code in the Content Authorization Response. The CSG detects the token, creates a new GET request that contains the original URL (without the appended token and any characters following it), and sends the GET on behalf of the client. The token must be known by the CSG, the quota server, and the notification server. The token is administratively defined on the CSG by using the CLI. The token must be chosen carefully to ensure that it is present only in URLs that are rewritten by the notification server and not in other client requests.

The CSG supports URL rewriting for HTTP, WAP 1.x, and WAP 2.x.

To define a URL-rewriting token for CSG, use one of the following commands in CSG user group configuration mode:

- **aoc confirmation**
- **verify confirmation**

WAP URL Appending

Whenever a Content Authorization Response contains a REDIRECT_URL action code for a WAP Content Authorization Request, the CSG can optionally append the originally requested URL to the one returned by the quota server.

For example, if the client requests the following URL:

http://www.redirect_url.com/home.wml

and the quota server returns the following URL in a REDIRECT_URL Content Authorization Response:

http://www.redirect_url.com/charges.wml

then the CSG sends the following URL as part of a redirect message to the client:

http://www.redirect_url.com/charges.wml?www.redirect_url.com/home.wml

The default behavior is to pass the redirect URL to the client as specified by the quota server without modification.

To enable WAP URL appending, set the CSG_WAP_APPEND_AOC_URL environment variable using the **variable** command in module CSG configuration mode.

SMTP Content Authorization

The CSG handles content authorization for SMTP slightly differently than it handles support for other protocols.

Typically, the CSG sends the Content Authorization Request immediately after performing service authorization. The CSG can do this because all of the information in the Content Authorization Request is contained in the initial flow received by the CSG.

However, for SMTP, the information needed in the Content Authorization Request—number of recipients with a good return code, number of recipients with a bad return code, size of e-mail in bytes (if present) and the sender of the e-mail—are not known until after the CSG processes the SMTP envelope. Therefore, when content authorization is configured, the CSG allows all envelope information to flow through, even if the user has no quota (however, access is not permitted if the user is not authorized). The CSG initiates the Content Authorization Request when it receives the DATA command. The CSG queues the packet containing the DATA command until content authorization processing is resolved.

If the CSG receives a DROP or REDIRECT in the Content Authorization Response, it drops the DATA command packet, terminates the session, and generates a CDR containing the envelope information and an invalid application return code.

If the CSG receives a FORWARD, it uses the weight that is returned in the response for prepaid processing.

Multiple e-mails over the same TCP connection result in multiple Content Authorization Requests. Each e-mail is treated as a separate transaction.

To enable content authorization for SMTP, use the **authorize content** command in CSG service configuration mode.

Redirect Flexibility

A quota server can request a redirect for multiple reasons (top-up, “sorry” indication, login request). The CSG allows the quota server to return the IP address and port number for each redirect. Thus, a different port number, or even a different server, can be used for every reason that the quota server might request the redirect. The CSG stores the most recent redirect address and port number for each service under each user profile, and uses that address and port instead of the globally defined default.

Service Verification

Service verification is a capability similar to advice of charge (AoC), which is provided the first time a user accesses a service using HTTP or WAP. A Service Verify Request quota management message supplies the quota server with content from the user request (the URL, header information, user agent, and so on). The quota server responds with a Service Verification Response that includes a decision to redirect the request to a notification server, to forward it, or to drop it.

Service verification provides the same URL-rewriting capabilities that are provided by AoC. An administrator uses the command-line interface (CLI) to define the service confirmation token that is used in URL-rewriting.

To enable or disable service verification, use the **verify** command in CSG service configuration mode. Service verification is also disabled when the quota server sends a Service Verify Tag-Length-Value (TLV) in a Service Authorization Response or Service Verification Response.

Service verification is supported only for HTTP and WAP.

As long as service verification is enabled, sessions of any type for this user do not trigger service reauthorization requests. Service reauthorization resumes for the user when service verification is disabled.

Service verification supports forward, redirect-URL, and drop authorization action codes sent in a Service Verification Response. Service verification also supports optional downloading of quota for a user in a Service Verification Response. The CSG sends service verification requests even when no quota is supplied in the Service Verification Response, if the Service Authorization Response contains the cause TLV with value 0x04 (user low on quota, but service access is permitted). Quota Download call detail records (CDRs) are sent to the BMA, as appropriate, whenever the quota server supplies quota in a Service Verification Response.

Service verification can be used in conjunction with existing AoC functionality.

CSG User Table

The CSG User Table identifies all users known to the CSG. The table is populated based on the contents of RADIUS Accounting Start messages, or from the user database, if either feature is enabled in your configuration. When the CSG processes a data packet, it searches the User Table and queries the user database for source and destination IP addresses that match the data packet:

1. If the User Table contains an entry with a matching source IP address, the CSG uses that entry for charging and for CDR generation.
2. If not, then if the User Table contains an entry with a matching destination IP address, the CSG uses that entry for charging and for CDR generation.

3. If not, then if the user database is configured and it contains an entry with a matching source IP address or destination IP address, the CSG uses the IP address that is returned (and the associated user ID) for charging and for CDR generation.
4. If neither the User Table nor the user database contains a matching source or destination IP address, then the CSG uses postpaid charging for the data packet, and the generated CDRs do not contain a user ID.

RADIUS Features

The CSG provides the following RADIUS features:

- [User Profile Retrieval from RADIUS Access-Accept or RADIUS Accounting Request, page 1-21](#)
- [Reporting RADIUS Attributes, page 1-21](#)
- [User Cleanup on RADIUS Accounting Start, page 1-22](#)
- [Processing Multiple RADIUS Accounting Stops, page 1-22](#)
- [RADIUS Monitor, page 1-23](#)
- [RADIUS Endpoint, page 1-23](#)
- [RADIUS Proxy, page 1-24](#)
- [RADIUS Handoff, page 1-25](#)
- [RADIUS Packet of Disconnect, page 1-25](#)
- [CSG Interface Awareness, page 1-26](#)

See the “[Configuring RADIUS Support: Learning Who the Subscriber Is](#)” section on page 5-1 for more information about configuring RADIUS features.

User Profile Retrieval from RADIUS Access-Accept or RADIUS Accounting Request

The user profile (billing plan) can be specified in a RADIUS message by using the Cisco subattribute 1 VSA. The format of the VSA is:

```
csg:billing_plan= billing_plan_name
```

The *billing_plan_name* can be null, to indicate a postpaid user. Otherwise, the billing plan name must be sent as an uppercase string to match a configured billing plan on the CSG.

The billing plan is included in the RADIUS Access-Accept message or RADIUS Accounting-Request message.

If the RADIUS Access-Accept message includes the billing plan, the user ID (RADIUS attribute 1 or 31, as configured) must also be included; otherwise, the CSG cannot associate the billing plan with the user.

Use the **user-profile server radius** command to retrieve the billing plan from the RADIUS message.

Reporting RADIUS Attributes

You can specify a set of attributes to be extracted from RADIUS Accounting Start messages for each subscriber, and to be reported with each transaction record. The CSG reports these attributes to the BMA and to the quota server. The CSG extracts these attributes from the RADIUS Accounting Start message, and refreshes (replaces) its stored attributes whenever a RADIUS Interim Accounting message is received, to ensure that the latest user information is stored.

You can use Arbitrary RADIUS attributes to determine where a user is connecting to the network, and for correlation purposes. Examples of these attributes and their uses include:

- NAS-IP-Address (4) identifies the gateway that provides accounting control for the subscriber. Examples of such devices include the gateway general packet radio service (GPRS) support node (GGSN), the Packet Data Serving Node (PDSN), the Home Agent, and the Cisco AS5300 Universal Access Server.
- SGSN IP (26/10415/6) identifies the Service GPRS Support Node (SGSN) that the subscriber is accessing, if the CSG is configured to report all incidents of RADIUS attribute 26 (the vendor-specific attribute, or VSA) instances.
- Acct-session-ID (44) uniquely identifies the session and can be used correlate GGSN accounting records. The CSG cannot separate each incidence of RADIUS attribute 26—it sends all of them.

The attributes are configured by their standard number, as shown in the following example:

```
ip csg accounting USER-BMA1
user-group GROUP1
agent activate 2 sticky 30
agent 210.0.0.102 3386 1
report radius attribute 3
report radius attribute 5
report radius attribute 7
report radius attribute 44
inservice
```

You can specify as many attributes as you want. If you specify so many attributes that the total message size is greater than a single UDP packet, the CSG supports continuation messages. A continuation message includes a correlator, a continuation number (so that messages that are received out of order can be reordered), and an indication of the final message.

If the configured attributes change, only the new RADIUS requests are subject to the new attributes. Attributes already saved for a user continue to be reported.

When a RADIUS Start Request is received, any attributes received from a previous Start Request are deleted. If there are multiple instances of an attribute, they are all reported. Attributes are reported in the same order that they are given in the RADIUS message.

User Cleanup on RADIUS Accounting Start

A subscriber's connectivity attributes might change over time without a RADIUS Accounting Stop message arriving to close down the previous accounting. Instead, it is possible that a new RADIUS Accounting Start message or a RADIUS Interim Accounting message might arrive with the updated information. Some customers might choose to close all of a user's services if a significant change has occurred in the user's status.

With the **radius start restart session-id** command configured, the CSG deletes the user entry as if it had received a stop, closes all of the subscriber's services, and creates a new entry.

To avoid deleting the user entry because of a retransmission of the RADIUS message, the **radius start restart session-id** command specifies an attribute to detect duplicate messages. If the contents of the attribute in the message match the contents of the previous message, the existing entry is not deleted.

Processing Multiple RADIUS Accounting Stops

For enhanced network connectivity options, such as secondary packet data protocol (PDP) contexts, the NAS sends multiple RADIUS Accounting Stop messages. In the case of secondary PDP contexts, for example, the NAS sends a RADIUS Accounting Stop as each context is terminated.

The CSG removes the subscriber from the User Table when it receives the final stop, which contains an attribute indicating it is final. The CSG support for this functionality allows the specific attribute to be configured. If this function is configured, the CSG processes only the RADIUS Accounting Stop that contains the configured attribute. The contents of the specified attribute are not examined.

**Note**

Retransmitted RADIUS Accounting Stop messages might cause problems when associating traffic with a subscriber. To avoid any problems, do not configure your RADIUS server to reuse an IP address immediately after it is released by a user.

RADIUS Monitor

RADIUS monitor provides a way to insert the CSG into the RADIUS flow without changing the authentication, authorization, and accounting (AAA) or Network Access Server (NAS) addresses in the network. The CSG monitors the traffic between the RADIUS client and the RADIUS server, and watches for RADIUS messages that match the configured rule. The address of the server must be configured.

Optionally, a RADIUS key is configured. If the key is configured, the CSG parses and acts on the message only if the RADIUS Authenticator is correct. If the key is not configured, the CSG always parses the message. The message is forwarded regardless of the configuration or accuracy of the key.

Here is a sample configuration:

```
ip csg user-group U1
  radius userid User-Name
  radius monitor 10.2.3.4 1234 key cisco --> Address, Port, and Key for RADIUS AAA Server.
  radius monitor 10.2.3.9 1234 key cisco2
  radius monitor 10.2.7.4 3901 key cisco --> Multiple AAA destinations can be monitored.
```

All RADIUS messages, including Access messages, are forwarded, except when the IP or UDP headers specify a length larger than the physical packet size.

All RADIUS messages, including access messages, are forwarded, except when the IP or UDP headers specify a length larger than the physical packet size.

When configuring RADIUS monitor for a server that is in the same subnet as a CSG interface, you must first configure a dummy route for that server, such as:

```
route ip-address 255.255.255.255 gateway gw-ip-address
```

where:

- *ip-address* is any IP address that is not used in the network
- *gw-ip-address* is the gateway IP address

Add a RADIUS monitor configuration only after you have added the dummy route.

RADIUS Endpoint

To configure the CSG as a RADIUS Accounting endpoint, and to use RADIUS PoD with a RADIUS endpoint, use the **radius endpoint** command in module CSG configuration mode.

To support RADIUS endpoint, the CSG requires a route to 255.255.255.255. You can configure the route by using the **gateway (module CSG VLAN)** command or the **route (module CSG VLAN)** command. For example:

```
gateway 31.0.0.6
```

or:

```
route 255.255.255.255 255.255.255.255 gateway 31.0.0.6
```

RADIUS Proxy

The CSG can act as a RADIUS proxy, forwarding all of the RADIUS Accounting messages it receives to a configured RADIUS server. When the RADIUS server acknowledges a message with an ACK, the CSG forwards the ACK to the client. RADIUS proxy supports both RADIUS Access and RADIUS Accounting.

The CSG allows you to specify only one RADIUS server, and the same RADIUS password must be used throughout.

RADIUS proxy can operate with clients that use many port numbers. The RADIUS client sends messages to the configured CSG (virtual) address. The CSG accepts messages for all ports on the configured address. The address of the RADIUS server is also configured. Optionally, a RADIUS key is configured. If the key is configured, the CSG parses and acts on the message only if the RADIUS Authenticator is correct. If the key is not configured, the CSG parses the message with no conditions. The message is forwarded regardless of the configuration or accuracy of the key.

All RADIUS messages (including Access messages) are forwarded except when the IP or UDP headers specify a length larger than the physical packet size.

To configure a CSG as a RADIUS proxy, use the **radius proxy** command in module CSG configuration mode. Keep in mind the following considerations:

- We recommend that you use RADIUS proxy support for only a small number (approximately 20) of RADIUS senders, where a sender is defined by its IP address and port.
- You can define up to 64,511 clients, when a client is defined by its IP address and port.
- The CSG IP address must be a virtual IP address, and it must be unique. The CSG IP address must not be specified in other CSG commands, and it must not match any real IP address, virtual IP address, or alias IP address configured on the CSG or in a /32 content configuration.
- You can configure a source IP address by using the **radius proxy** command in module CSG configuration mode. The CSG uses the source IP address when it forwards a RADIUS message to the server.
- If you are load-balancing more than one CSG, you must configure the CSG's as RADIUS proxies, not RADIUS monitors.

The CSG RADIUS interface recognizes the following Cisco-specific VSAs:

- Subattribute value **csg:quota_server=<ip>:<port>** includes the quota server IP address and port in a RADIUS Start Accounting Message. You must manually configure the quota server referenced by this subattribute in order for the CSG to act on this VSA. If the quota server is not configured, the CSG creates a null entry in the User Table for the quota server. The user specified by the RADIUS message uses the quota server in the VSA.
- Subattribute value **csg:downlink_nexthop=<ip>** includes the downlink next-hop IP address in a RADIUS Start Accounting Message. The downlink next-hop IP address is the address to which all downlink traffic is sent for a given user IP address, plus table pairing. If this VSA is not present, traffic is routed based on the routing tables of the CSG.

For RADIUS endpoint and RADIUS proxy configurations, you can prevent the CSG from acknowledging the following errors by entering the **no** form of the **radius ack error** command in CSG user group configuration mode:

1. The User Table entry cannot be created because of resource constraints.

2. The CSG parses the RADIUS Accounting Request and encounters RADIUS protocol errors.
3. The CSG parses the RADIUS Accounting Request and a billing plan is specified in the RADIUS Accounting Request, but it does not match a billing plan in the CSG configuration.
4. The CSG parses the RADIUS Accounting Request and a quota server is specified in the RADIUS Accounting Request, but it does not match a quota server in the CSG configuration.
5. The CSG parses the RADIUS Accounting Request and a connect service is specified in the RADIUS Accounting Request, but it does not match a connect service in the CSG configuration.

For errors 3, 4, and 5, the CSG can parse the configuration VSA from the RADIUS Access-Accept. If the CSG uses any attribute from the RADIUS Access-Accept that does not match the CSG configuration, the CSG does not send a RADIUS response to the RADIUS Accounting Request.

For RADIUS Accounting requests processed as a result of matching a **radius endpoint** command, the CSG does not send a RADIUS acknowledgement.

For RADIUS Accounting requests processed as a result of matching a **radius proxy** command, the CSG does not forward the RADIUS Accounting Request to the RADIUS server.

RADIUS Handoff

In networks that do not use Cisco home agents (HAs), the CSG's RADIUS handoff feature can manage handoffs for roaming users.

When RADIUS handoff is configured, and a RADIUS Accounting Stop is received, the CSG starts a handoff timer instead of immediately deleting the CSG User Table entry for the roaming user.

- When a handoff occurs, the CSG detects a RADIUS Accounting Start message for the same user with a different network address server (NAS) IP address. The CSG then uses the existing User Table entry for the user, to preserve the user information, and turns off the timer.
- If the handoff timer expires before the CSG detects a RADIUS Accounting Start message for the user, the CSG assumes a handoff did not occur and deletes the User Table entry for the user.
- In the event of a failover, all handoff timers are restarted.

To configure RADIUS handoff support, use the **radius handoff** command in CSG user group configuration mode.

RADIUS Packet of Disconnect

The quota server can instruct the CSG to disconnect a user. The CSG then sends a RADIUS Packet of Disconnect (PoD) message, identifying the user, to the NAS, and the NAS sends a RADIUS Accounting Stop message, which also clears the User Table entry.

By using one of the following methods, the quota server instructs the CSG to disconnect a user:

- The quota server can send the UserDisconnectRequest message to the CSG. This message uses the UserIndex TLV to identify the user to be disconnected.
- The quota server can use Action Code 4 in the Action TLV in one of the following requests and responses:
 - The Service Authorization Response (indicating that the CSG will send the PoD message when the quota runs out)
 - The Service Stop Request (indicating that the CSG will send the PoD message immediately)
 - The User Profile Response (indicating that the CSG will send the PoD message immediately)

The CSG sends the PoD message to the NAS that is specified by the NAS-IP-Address attribute (4) in the RADIUS Accounting Start.

To configure RADIUS PoD support, use the following commands in CSG user group configuration mode:

- Use the **radius pod attribute** command to specify the RADIUS attributes to be copied from the RADIUS Start message and sent to the NAS in the PoD message.
- Use the **radius pod nas** command to specify the NAS port to which the CSG must send the PoD message, and the key to use in calculating the Authenticator.
- Use the **radius pod timeout** command to specify the number of times to retry the RADIUS PoD message if it is not acknowledged, and the interval between retries.

The CSG can send PoD messages only if the CSG received the user information on a virtual interface that is configured by using the **radius proxy** or **radius endpoint** command in module CSG configuration mode.

CSG Interface Awareness

Many provider networks offer data access, control over user addressing, and dedicated Virtual Routing and Forwarding (VRF) over the wireless network to enterprises and Mobile Virtual Network Operators (MVNOs). Interface awareness enables the CSG to distinguish between users and sessions that share the same IP address on different VLANs (that is, users and sessions with overlapping IP addresses).

Because the quota server can only respond to the CSG (that is, there can be no quota server-initiated messages), the Extended User Index TLV is required in order to identify or trigger action for a user within a CSG table.

The CSG binds contents to specific VLANs, and configures those VLANs with a table ID. When user packets match a content and a session is created, the CSG uses the table ID of the content as part of the user search. When a table is configured on a VLAN, the contents that are bound to each VLAN are associated with the table ID of that VLAN.

To support traffic segregation across VLANs, the CSG uses next-hop to bind flows to uplink and downlink routing hops. The CSG routes uplink packets (from the Network Access Server [NAS]) by applying next-hop policies to the contents on each NAS VLAN. The CSG routes downlink packets via the downlink address supplied by the NAS in the RADIUS Accounting start message.

Logically, that means that a dedicated per-VLAN NAS is required for interface awareness. Physically, however, it depends on the capabilities of the NAS. Each RADIUS proxy statement can have a table name. When a CSG User Table entry is created as a result of a Start message sent to that proxy IP address, the specified table name is associated with the user. Depending on your network, you might choose to route this user's traffic different from another user's traffic, even when the source or destination IP addresses are the same. To do so, use the next-hop command in CSG policy configuration mode, or specify the downlink next-hop in the Start message.

To associate a table name with a VLAN, use the **table** command in module CSG VLAN configuration mode.

To associate a table name with a particular RADIUS proxy or endpoint, use the **radius proxy** and **radius endpoint** commands in module CSG configuration mode.

HTTP Features

The CSG provides the following HTTP features:

- [HTTP Pipelining and Chunked Transfer Encoding, page 1-27](#)
- [HTTP 1.0 Content Billing, page 1-27](#)
- [HTTP 1.1 Content Billing, page 1-27](#)
- [HTTP Records Reporting Flexibility, page 1-27](#)
- [HTTP Error Code Reporting, page 1-28](#)
- [Intermediate Billing Records, page 1-52](#)

HTTP Pipelining and Chunked Transfer Encoding

The CSG supports full HTTP pipelining and chunked transfer encoding.

If pipelined connections are replicated to a standby CSG, and a failover occurs, the CSG does not increment the content counters for traffic flowing through these connections. The CSG does increment the content counters for new pipelined connections created after the failover.

When performing AoC for a TCP connection carrying pipelined HTTP requests, the CSG responds with the redirect to the client as soon as the quota server requests the redirect. This could result in the redirect arriving at the client before responses for previous requests arrive, and the client might associate the redirect with a different request in the pipeline.

There are no commands required to enable this function.

HTTP 1.0 Content Billing

The CSG enables you to bill users for individual transactions by discriminating on a per-object basis, and on a per-user basis. Unlike traditional billing models, which bill for broad classes of traffic, this service enables differentiated billing based on the actual object being requested. You can even bill objects at different rates to different customers. For example, you can bill advertisements to the advertiser, rather than to the end user.

HTTP 1.1 Content Billing

The CSG separately records each request over a persistent HTTP 1.1 session.

HTTP Records Reporting Flexibility

The client's IP address is included in the HTTP Header message. This enables the BMA to identify the client by user ID (and by IP address) immediately, without having to wait for the HTTP Statistics record.

You can configure the CSG to send the HTTP Header message as soon as it is generated, rather than holding it until an entire packet is filled. This reduces latency and notifies the BMA about the client's transaction as quickly as possible. Although this type of reporting is more efficient, it provides less information; use it only when the BMA needs to react to the client's activity very quickly.

You can configure the CSG to not send the HTTP Statistics message. This configuration reduces the load on the BMA and is useful when the billing policy depends only on the event and does not require detailed statistics. Note that the CSG still sends the HTTP Statistics message if the session fails (for example, if a Reset [RST] is received without a Finish [FIN], or if the session times out).

HTTP Error Code Reporting

The CSG reports HTTP-specific information about the request, such as the URL, as well as HTTP error codes (response codes of 300 or higher).

WAP Features

The CSG provides the following WAP features:

- [WAP Traffic, page 1-28](#)
- [WAP 2.0, page 1-28](#)
- [WAP Cutoff, page 1-29](#)

WAP Traffic

The CSG can intercept WAP traffic and generate reports that include contextual WAP information and counts of the bytes transferred. WAP functionality provides protocol-level prepaid and postpaid billing, including the following functionality:

- Billing CDRs for Wireless Transaction Protocol (WTP) and WSP in support of WAP 1.2—The ability to generate billing records for each WAP GET, POST, PUSH or CONFIRMED PUSH, ABORT and REPLY PDUs, as well as a summary report at WAP Disconnect. Records include URL, User Agent, source and destination IP, separate IP byte and PDU counts from both the initiator and the responder. (The PDU count is not the same as the packet count. Multiple WAP PDUs can share a single packet.)
- Prepaid billing for WTP and WSP in support of WAP 1.2, including the ability to differentiate WAP browsing from the Multimedia Messaging Service (MMS), and to exclude charging for MMS.
- Top-up capability using URL redirect.
- URL-map support for WAP.
- Support for multiple services.
- WAP 1.2.1 HTTP support: The CSG HTTP support is compatible with WAP 1.2.1 (HTTP over WP-TCP) traffic.

WAP 2.0

The CSG supports billing for the following types of WAP 2.0 network flows:

- Retrieving a message from the network using HTTP.request-method: GET
- Posting a message into the network using HTTP.request-method: POST
- Acknowledging a PUSH indication using HTTP.request-method: POST

WAP 2.0 mobile devices can participate in these flows, implemented as WAP 2.0/HTTP/TCP across a WAP 2.0 Proxy or Push Proxy Gateway (PPG): WAP 2.0 mobile devices can be configured to use the WAP 2.0 proxy or to ignore it. However, if a WAP 2.0 proxy is not configured, the configuration resembles HTML over HTTP (in that you must choose the appropriate content rules so that HTTP policies can be applied to the WAP 2.0 traffic). The WAP 2.0 proxy enables you to identify WAP 2.0 traffic by configuring a content that examines traffic to and from the WAP 2.0 proxy. Using an account

type of **http** enables billing of WAP 2.0, including support for policies based on the HTTP method, URL and HTTP header values. The current limitations of HTTP billing (with respect to Transport Layer Security [TLS]) apply to billing WAP 2.0/HTTP and MMS/WAP 2.0/HTTP.

The CSG also supports PPG-Originated TCP (PO-TCP), implemented as WAP 2.0 over SMS OTA-PUSH rather than WAP 2.0 over HTTP. In PO-TCP, the PPG establishes a direct connection to the mobile device using prior knowledge of its IP address. The PPG negotiates an understanding of the mobile device identity and capabilities by using HTTP.request-method: OPTIONS, and then uses HTTP.request-method: POST to deliver the PUSH notification as a WAP 2.0 XML message.

WAP 2.0 mobile devices can implement support for extensive MMS over WAP 2.0. Service providers use MMS to differentiate and promote their products; thus, the billing of MMS over WAP 2.0 needs to be differentiated from other WAP 2.0 billing. The CSG can bill MMS over the supported WAP 2.0 flows at a differentiated rate by using the capabilities of the **http** accounting type to detect some or all of the following characteristics of MMS/WAP 2.0/HTTP traffic:

- The URL of a GET of MMS content points to the MMSC and encodes an MMS message ID.
- The URL of the POST of an MMS message or an MMS message notification acknowledgement points to the MMSC.
- The Content-Type HTTP header of the POST of an MMS message or an MMS message notification acknowledgement is “application/vnd.wap.mms-message”.

MMS over WAP 2.0 allows the following types of notification:

- PO-TCP
- SMS-based notification carrying the Uniform Resource Identifier (URI) for the MMS. The handset then initiates a GET request to that URI to retrieve the information.
- TO-TCP (Terminal-Originated TCP), which starts with SMS, but provides only the IP address of the PPG. The handset must then open a TCP connection and wait for an HTTP request from the PPG. This HTTP request is an OPTIONS method. It must succeed before the handset can retrieve the notification.

The CSG Layer 7 billing for MMS relies entirely on the PO-TCP and SMS-based notification types. TO-TCP is not supported.

**Note**

If a terminal reuses a persistent PO-TCP to initiate a new method request, the packets are dropped and the PO-TCP connection appears hung until TCP retry attempts expire.

WAP Cutoff

When a user’s quota is depleted in the middle of a WAP transaction, the CSG allows the transaction to complete and charges the user for all bytes used, resulting in a negative quota balance for the user. On the next transaction request, the CSG redirects the user to the top-off server. This method of operation provides the best experience for the user, but it also allows for some leakage of quota. For small transactions, the leakage is minimal; however, for large transactions the leakage can be significant.

The CSG enables you to balance user experience versus quota leakage on a per-service basis by using the **zero-quota abort type** command in global configuration mode. This balancing is supported only for WAP; the default is to allow a transaction to continue when the user runs out of quota midstream.

**Note**

Configuring the cutoff option for WAP affects only connection-oriented sessions, and not connectionless traffic.

When configured, WAP cutoff causes the CSG to end an existing transaction when the user runs out of quota. The CSG sends termination messages to both the client and server, ending the transaction. A BMA record for the transaction is generated with a flag setting in the Wireless Transaction Protocol (WTP) information record that indicates the transaction was intentionally ended. In the report, the user is charged for the number of bytes that were processed for the transaction, including the bytes that caused it to exceed the quota balance. Typically, the user is not charged for this transaction because the transaction did not complete. The user is reimbursed by the billing agent for transactions with the 0x04 flag set, or is reimbursed by the prepaid refund feature.

RTSP Features

The CSG provides the following Real Time Streaming Protocol (RTSP) features:

- [RTSP Billing, page 1-30](#)
- [Per-Click Authorization, page 1-30](#)
- [Correlation, page 1-31](#)

RTSP Billing

The RTSP Billing feature adds the following functionality to the CSG:

- Correlation of various streams associated with an RTSP session
- Reporting of application-level information (for example, filename) to the billing system

RTSP uses the following protocols for streaming to the client. The client presents the server with a choice of acceptable protocols and port numbers, and the server responds with its choice of protocol that includes:

- RTSP requires a control TCP connection to server port 554.
- RTSP also requires a UDP server-to-client stream for RTP (audio/video stream delivery), and a bidirectional UDP flow pair for exchanging synchronization information. The ports for the UDP flows are negotiated on the TCP connection during the SETUP exchange.
- RTSP can use RealNetworks Data Transport (RDT) for the stream transport. This establishes a UDP flow in each direction: one for stream delivery from the server, and the other for requesting the resending of lost media packets.
- RTSP can operate completely over the single TCP connection.
- RTSP can be tunneled over HTTP.

RTSP transport modes are negotiated on the control connection using the following methods:

- The client sends a SETUP request that identifies one or more modes it can support.
- The server responds with a mode that it has selected and ports that are to be used.

Per-Click Authorization

Per-click authorization implements functions like AoC redirection and retrieval of price from an external server. For the control session, the CSG sends a Content Authorization Request at the beginning of the TCP session. For each transaction involving a data stream, the CSG sends a Content Authorization Request before it allows the data stream to flow. This request allows the quota server to inspect the filename before granting authorization.

The CSG allows only Network Address Translation (NAT) redirection for RTSP traffic.

RTSP allows the multiplexing of multiple data streams over the same transport. For example, audio and video presentations can be multiplexed over the same UDP flows. The quota server must ensure that it does not send contradictory responses to the various Content Authorization Requests. For example, if one request is allowed and the other one is denied, the behavior of the CSG is undefined.

Correlation

The CSG provides RTSP correlation at the RTSP session level. All TCP/UDP flows associated with an RTSP session share a correlator.

The CSG does not correlate RTSP streams that do not share the RTSP session ID.

Correlating Multiple Streams Controlled by a Single RTSP Session

An RTSP session can control multiple streams, such as the audio stream and the video stream for a movie. For instance, a client can perform the following operations over the same RTSP session:

- DESCRIBE rtsp://a.ex.com/movie.sdp

The client requests the description of a movie. The server assigns a session ID to the client, and sends the.sdp file containing information about the movie.

- SETUP rtsp://a.ex.com/movie/audio

The client requests the setup of a stream.

- SETUP rtsp://a.ex.com/movie/video

The client requests the setup of a second stream. This results in the setting up of four UDP flows.

- PLAY rtsp://a.ex.com/movie.sdp

In this example, all the streams share the RTSP session and the session ID. There is one RTSP control TCP session, with four associated UDP streams. The CSG correlates all four UDP streams with the control session.

Correlating Multiple Streams Controlled by HTTP

HTTP sessions can be used to correlate multiple related RTSP streams. Different RTSP streams could go to different servers. The CSG has no easy way to determine which two streams are related. For example, a web server (W) hosts the media description file, movie.sdp; a video server (V) contains the video stream; and an audio server (A) contains the audio stream. [Table 1-1](#) identifies the interactions that occur.

Table 1-1 Multiple Streams Controlled by HTTP

Client	Server	Protocol	Method/URL
C	M	HTTP	GET /movie.sdp
C	V	RTSP	SETUP rtsp://v.eg.com/video
C	A	RTSP	SETUP rtsp://a.eg.com/audio
C	V	RTSP	PLAY rtsp://v.eg.com/video
C	A	RTSP	PLAY rtsp://a.eg.com/audio

In the previous example, there are five concurrent sessions:

- One HTTP 1.1 session

- Two RTSP video sessions
- Two RTSP audio sessions

All of the TCP and UDP sessions associated with an RTSP session can be correlated. In this same example, the sessions associated with the video on server V are correlated. Similarly, the sessions associated with the audio on server A are correlated; however, there is no correlation between the audio and video flows, and neither the audio flow nor the video flow is correlated with the HTTP session.

Implications of Container Files

A container file is a storage entity in which multiple, continuous media types pertaining to the same end-user presentation are present. A container file represents an RTSP presentation; each of its components is an RTSP stream. While the components are transported as independent streams, it is desirable to maintain a common context for these streams at the server. Synchronized Multimedia Integration Language (SMIL) is an example of a programming language that can be used to describe the contents of a container file.

The CSG does not correlate the streams within a container file.

Interleaved RTSP

Interleaved RTSP passes RTSP data in the TCP control session. Because the CSG parses the control session, it could cause a large performance bottleneck.

To avoid bottlenecks, the CSG performs the following actions for interleaved RTSP sessions:

- Waits for a SETUP request/reply to determine whether this is an interleaved RTSP session.
- Remembers the URL information.
- After determining interleaved RTSP, reports RTSP information to the BMA/quota server, and begins fastpath processing for the connection. Any subsequent transactions on the same RTSP control connection are not visible to the CSG's billing function.

This method provides some RTSP-level information, but avoids making the RTSP path a target of denial-of-service (DoS) attacks. If most of the RTSP streaming billing applications are protected, customers have some control over the servers to ensure that interleaved RTSP is not used excessively.

CDRs

The CSG generates the following CDRs for RTSP:

- TCP control session: TCP, TCPInt, RTSP
- Data streams: RTSP stream
- UDP CDRs for each UDP session



Note If you are using fixed CDR support, the CSG does not generate any UDP CDRs.

RTSP billing in the CSG is based on inspection of the RTSP SETUP and TEARDOWN messages that are exchanged between the client and server. The CSG builds the RTSP CDR immediately after the RTSP TEARDOWN signal if the URL exactly matches the URL from the RTSP SETUP signal. Otherwise, the CSG builds the CDR after any condition that causes the flows to be terminated, such as:

- When the content idle timer expires. By default, this timer is set to 3600 seconds (1 hour). To receive the RTSP CDRs sooner, set the timer to a smaller value, such as 60 seconds, using the **idle** command in CSG content configuration mode.
- When a service_stop is triggered (for example, when the access server sends a RADIUS Accounting Stop for the user).

Session Processing

RTSP control session processing is similar to FTP control session processing. An 8-byte correlator is assigned to the RTSP control session. The most significant 6 bytes of the correlator are assigned from the session ID and the session ID sequence. The least significant 2 bytes of the correlator are zeroed (for example, 0x0000).

The CSG keeps track of RTSP sessions, and an RTSP session is used to correlate multiple streams that are associated with the session.



Note

An RTSP session might comprise more than one TCP session; alternatively, an RTSP session can exist without even one TCP session between client and server.

When the client sends a **setup** command, the CSG notes the client ports, and extracts the server port information from the SETUP response. Data connections to these ports are processed as if they hit the *content, policy* definition for the control server.

The following example (from RFC 2326) uses a single RTSP session to control multiple streams. The CSG actions are annotated after various steps.

In this example, the client (C) requests a presentation from the media server (M). The movie is stored in a container file. The client has attached an RTSP URL to the container file.

```
C->M: SYN port=RTSP
M->C: SYN-ACK
Assign 8 byte correlator X. Lower two bytes of the correlator are 0.

C->M: DESCRIBE rtsp://foo/twister RTSP/1.0
      CSeq: 1

M->C: RTSP/1.0 200 OK
      CSeq: 1
      Content-Type: application/sdp
      Content-Length: 164

      v=0
      o=- 2890844256 2890842807 IN IP4 172.16.2.93
      s=RTSP Session
      i=An Example of RTSP Session Usage
      a=control:rtsp://foo/twister
      t=0 0
      m=audio 0 RTP/AVP 0
      a=control:rtsp://foo/twister/audio
      m=video 0 RTP/AVP 26
      a=control:rtsp://foo/twister/video

C->M: SETUP rtsp://foo/twister/audio RTSP/1.0
      CSeq: 2
      Transport: RTP/AVP;unicast;client_port=8000-8001

M->C: RTSP/1.0 200 OK
      CSeq: 2
      Transport: RTP/AVP;unicast;client_port=8000-8001;
                server_port=9000-9001
      Session: 12345678
```

Build RTSP record. Correlator = X + i. The CSG makes sure that X + i results in an even number. RTSP usage records for these two UDP flows carry X + i and X + i + 1 as the correlators. The correlators share 63 bits to help bind together the various flows for an RTSP transaction; that also enables you to distinguish the various interim records for one UDP flow from another.

```

C->M: SETUP rtsp://foo/twister/video RTSP/1.0
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8002-8003
      Session: 12345678

M->C: RTSP/1.0 200 OK
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8002-8003;
                server_port=9004-9005
      Session: 12345678

```

Build RTSP record. Correlator = X + 3. RTSP usage records generated for these two UDP flows carry the same correlator.

```

C->M: PLAY rtsp://foo/twister RTSP/1.0
      CSeq: 4
      Range: npt=0-
      Session: 12345678

M->C: RTSP/1.0 200 OK
      CSeq: 4
      Session: 12345678
      RTP-Info: url=rtsp://foo/twister/video;
                seq=9810092;rtptime=3450012

C->M: TEARDOWN rtsp://foo/twister RTSP/1.0
      CSeq: 6
      Session: 12345678

V->C: RTSP/1.0 200 OK
      CSeq: 6

```

This TEARDOWN does not correspond to the SETUP URL; therefore, the CSG lets the streams idle out and sends the usage records when the streams idle out.

IMAP Support

The CSG provides the following Internet Message Access Protocol (IMAP) features:

- [Prepaid Support for IMAP, page 1-34](#)
- [Transaction Support for IMAP, page 1-34](#)

Prepaid Support for IMAP

The CSG supports prepaid billing for IMAP.

There are no commands required to enable prepaid support for IMAP.

Transaction Support for IMAP

The CSG provides transaction support for IMAP. The CSG defines an IMAP transaction as a tagged response from an IMAP server that contains TEXT. TEXT is the part of the e-mail that follows the envelope; the presence of TEXT results in a classification of BODY. The CSG includes IMAP transaction counts in the Completed Transactions TLV. The CSG does not include any envelope information in the IMAP transaction CDRs.

For requests and responses that are not transactions (they do not contain TEXT), the CSG accumulates the bytes and includes them in the next transaction. When the IMAP session ends, the CSG reports any remaining bytes.

Consider the following simple example of an IMAP transaction with BODY:

Client request: 1 FETCH 5 BODY[]

Server response: * 5 FETCH (BODY[]{55}**cr-lf-55-bytes-of-e-mail-followed-by-cr-lf)cr-lf
1 OK FETCH COMPLETE**

The CSG handles this request and response as follows:

-
- Step 1** The client request is tagged **1**. The CSG parses the request and increments the body up byte counts, because the request was for a **BODY[]**.
 - Step 2** The CSG parses the untagged response from the server and notes that it contains TEXT (**BODY[]**).
 - Step 3** The CSG parses the tagged response **1 OK FETCH COMPLETE**, which means this is an IMAP transaction (a tagged response that contains TEXT).
-

Here is a more complicated example:

Client request: 8 FETCH 1:100 BODY[]<0.5>

Server response: * 1 FETCH (BODY[]<0> “.”)cr-lf

*** 2 FETCH (BODY[]<0> “.”)cr-lf**

*** 3 FETCH (BODY[]<0> “.”)cr-lf**

*** 4 FETCH (BODY[]<0> “.”)cr-lf**

...

*** 100 FETCH (BODY[]<0> “.”)cr-lf**

8 OK FETCH COMPLETE

The CSG handles this request and response as follows:

-
- Step 1** The client request is tagged **8**. The CSG parses the request and increments the body up byte counts, because the request was for a **BODY[]**.
 - Step 2** The server sends 100 untagged responses which the CSG parses, noting that the response contains TEXT (**BODY[]**).
 - Step 3** The CSG parses the tagged response **8 OK FETCH COMPLETE**, which means this is an IMAP transaction (a tagged response that contains TEXT). The CDR reports 100 **BODY** fetches, the request bytes are allocated to body up, and the response bytes are allocated to body down.
-

The CSG categorizes bytes as BODY, HEADER, and OTHER, determined as follows:

- **BODY**—The bytes are classified as BODY if a fetch request or response is encountered for one of the following specifications (including any appended “<>” subset variants):
 - BODY[]
 - BODY[#]
 - BODY[TEXT]
 - BODY[#.TEXT]
 - BODY.PEEK[]
 - BODY.PEEK[#]

- BODY.PEEK[TEXT]
- BODY.PEEK[#.TEXT]
- RFC822
- RFC822.TEXT
- HEADER—If the bytes cannot be classified as BODY, then they are classified as HEADER if a fetch request or response is encountered for one of the following specifications (including any appended “<>” subset variants):
 - BODY[HEADER]
 - BODY[#.HEADER]
 - BODY.PEEK[HEADER]
 - BODY.PEEK[#.HEADER]
 - RFC822.HEADER
- OTHER—If request or response cannot be classified as BODY or HEADER, then it is classified as OTHER. OTHER examples include:
 - SYN/FIN/ACK/RST packets that do not contain a payload
 - Non-HEADER or BODY IMAP commands such as **3 select inbox**
 - Retransmitted packets
 - Anything else that is not considered BODY or HEADER
 - If the session becomes encrypted or enters PASSTHRU mode, subsequent packets for the session cannot be parsed and are treated as OTHER.

To specify which IMAP bytes are billed for when doing prepaid debits (BODY only, BODY and HEADER only, or BODY and OTHER only), use the **meter imap** command in CSG service configuration mode.

Because IMAP metering is byte-based, you cannot configure both **meter imap** and **basis fixed** or **basis second** in the same service. Only **basis byte** is meaningful with **meter imap**.

If you specify **basis fixed**, each IMAP transaction counts as a quadran, subject to weights.

To specify that the CSG is to refund quota for IMAP quota for application return codes, use the **retcode** command in CSG refund configuration mode.


Note

Any IMAP transaction that is not an OK tagged response (such as **1 OK FETCH COMPLETE**) is subject to a prepaid refund.

POP3 Support

The CSG generates a single CDR for each POP3 e-mail. The CDR includes all necessary information, such as the IP byte count and the TCP byte count. The CSG no longer generates a final TCP Stats record. If a user downloads multiple e-mails during a single TCP session, the CSG generates a CDR for the previous e-mail each time it processes a new POP3 RETR (retrieve message) or TOP (retrieve specific lines of the message body) command. The CSG generates a CDR for the last e-mail when it processes the POP3 STATS (statistics) command (for TCP termination).

The CSG supports POP3 in both prepaid and postpaid mode. For **basis fixed** prepaid billing, the CSG treats each e-mail download as a transaction and a prepaid debit subject to weighting.

The CSG also supports refunding for POP3. If an e-mail download request (TOP or RETR) flows from the client to the server, and the next server response is not **OK**, or the session ends without seeing **OK**, then the prepaid debit returns the prepaid quota consumed for this transaction. The refund return code used is 554; if you want the CSG to provide refunding for POP3, specify return code **554** on the **retcode** command in CSG refund configuration mode for the POP3 refund definition.

If a user tries to download e-mail and no e-mail exists, the CSG generates a POP3 CDR that contains an application return code TLV with a value of 554. This is the only condition in which the CSG includes a non-zero return code in a POP3 CDR.

To define the POP3 accounting type for a billing policy, use the **accounting type pop3** command in CSG policy configuration mode.

SMTP and POP3 Billing

SMTP is the Internet e-mail transfer protocol that operates over TCP with port 25. End users send messages using SMTP. SMTP is also used to transfer messages between SMTP gateways (or relays). POP3 is a common protocol used to retrieve Internet e-mail from an e-mail server. POP3 also operates over TCP and typically uses port 110.

SMTP and POP3 messages consist of the following parts:

- Envelope—The SMTP and POP3 commands and responses.
- Headers—RFC 2822 headers that appear as contents to the SMTP and POP3 protocols. The RFC 2822 headers are of the form “header field name: header field body.” Some common header field names are “To,” “From,” “Date,” “Subject,” “Cc,” and “Bcc.”
- Body—The part of the message that appears as contents to the SMTP and POP3 protocols, but does not include headers. The headers and body of the message are separated by a blank line (for example, <CR><LF><CR><LF> in RFC 2822).

The CSG inspects SMTP and POP3 messages and reports all RFC 2822 header field names and bodies that appear in the header section of the message (before the body of the message). SMTP and POP3 envelope information is not reported, except for the SMTP return code from the DATA command. For SMTP, the sender and recipients in the SMTP MAIL and RCPT commands are not reported, but the values from the “To,” “From,” “Date,” “Cc,” and “Bcc” headers in the contents of the e-mail message are reported to identify senders and recipients.

Because the amount of information in the header section might be greater than an IP packet encapsulated in an Ethernet frame, the information might span multiple records by using the CSG Continue Data Record type. Because the amount of information in a single header field might also be greater than an IP packet over Ethernet, the CSG Report String Attribute reports also has a continuation option. This means that information for a single header might span multiple CSG Report String Attribute reports, which might span multiple CSG Data Records.



Note

If a TCP connection carries multiple e-mail messages, each e-mail message generates a separate SMTP or POP3 Data Record (plus Continuation Data Records if necessary).

FTP Billing

The CSG supports both postpaid and prepaid FTP protocol-aware billing. The CSG can generate TCP billing records for FTP connections, and records that report FTP-specific information, such as the filename.

Users can define **basis fixed** and **basis byte** prepaid billing services for FTP.

**Note**

There is no regular expression (map) support for differentiating FTP services.

FTP requires a control TCP connection to well-known server port 21.

Header Mapping and URL Mapping

The CSG uses maps to match headers or URLs against a pattern, to determine whether flows will be processed by the CSG accounting services.

- The CSG provides header mapping and filtering for HTTP. For more information about header mapping, see the description of the **match (header map)** command.
- The CSG provides URL mapping and filtering for HTTP, RTSP, and WAP. For more information about URL mapping, see the description of the **match (URL map)** command.

Passthrough Mode and the Default Quota

For prepaid users, sessions are blocked when a quota server is not available for authorization grant of quota. In passthrough mode, the CSG grants quota for services and their sessions when a quota server is not available. The CSG allows all traffic to pass, and CDRs are flagged for special consideration by the BMA.

For each service for which you want to use passthrough mode, you must enable it using the **passthrough** command in CSG service configuration mode.

If you enable passthrough mode for a service, do not disable quota server reassignment for user groups associated with that service. That is, do not configure the **no quota server reassign** command in CSG user group configuration mode for user groups associated with the service.

You also use this command to specify the size of each quota grant (the default quota) to assign to a service. When passthrough mode is enabled for a service, and a session for a service needs quota, and no quota server is active, the CSG grants the service the amount of quadrans specified on the **passthrough** command. (There are three types of quadrans: **basis byte** for volume-based billing, **basis fixed** for event-based billing, and **basis second** for duration-based billing.) The CSG continues to grant quota as long as a quota server is inactive.

When the service becomes idle, the CSG generates and stores a Service Stop Request message, containing the total usage for this instance of the service. When a quota server becomes active, the CSG forwards all stored Service Stop Request messages to the quota server.

This section contains the following information about passthrough mode:

- [Flagging of Messages, page 1-40](#)
- [User Profile Requests, page 1-40](#)
- [Quota Server Recovery, page 1-40](#)

Flagging of Messages

To facilitate billing recovery, some messages to the quota server and the BMA include a QuotaServerFlags TLV. The CSG adds this TLV whenever it grants a passthrough mode quota to a service.

User Profile Requests

When the CSG learns of new users, it typically sends a User Profile Request to an active quota server. This enables the CSG to learn the billing plan to use for each user. If the quota server returns a NULL billing plan, this indicates that a user is postpaid.

When passthrough mode is in use for any service, the CSG changes the way it processes User Profile Requests. When there is no active quota server, the CSG assigns all new users to postpaid processing. The CSG reports all sessions for these users as postpaid, and does not flag generated CDRs with a QuotaServerFlags TLV.

If a user is still on the network when the quota server becomes active, the CSG sends a normal User Profile Request to the quota server for the user. When the CSG receives a response, it updates the user's billing plan. If the updated billing plan is now a prepaid billing plan, the CSG blocks new IP sessions started by the user until the quota server grants a quota. IP sessions that were active before the billing plan was updated to prepaid are kept as postpaid, and generate postpaid CDRs until they end.

Quota Server Recovery

When a quota server becomes active, the CSG forwards stored Service Stop Requests to it. Additional actions taken by the CSG depend on user traffic.

When a user who was forced to postpaid status while the quota server was absent creates a new IP session, the CSG issues a User Profile Request followed by a Service Authorization Request, and blocks new traffic until quota has been granted.

Prepaid users might have some services that were granted quota in passthrough mode. For those services, when quota runs low, the CSG sends a Service Reauthorization Request to the quota server, flagging the request with the QuotaServerFlags TLV. The usage TLV and remaining TLV contain the sum total of quota granted to the service since it began. This total might be a combination of quota granted by the quota server before the failure and quota granted by the CSG in passthrough mode. The requested quadrans TLV contains a request for an additional quota amount.

When the quota server responds to a Service Stop or a Service Reauthorization Request, the CSG moves the service out of passthrough mode. If the quota server denies quota when it sends a Service Authorization Response message, the CSG blocks the traffic. The CSG also flags CDRs generated by traffic for these services, which received passthrough mode quota grants, with QuotaServerFlags TLVs, until a Service Stop Request is sent. That is, once a service is granted a passthrough mode quota, the CSG flags all CDRs for that serviced, up to and including the Service Stop. This applies only to prepaid users. Postpaid users CDRs are never flagged.

Service Duration Billing

The Service Duration Billing feature enables the CSG to deduct quota based on the time of network usage for prepaid (or “managed”) users. With this feature, the user is charged for the time duration of the CSG service. The following sections describe how this feature works:

- [Charging for Service Duration Billing, page 1-41](#)
- [Calculating Usage for Service Duration Billing, page 1-41](#)
- [Reporting Quadrans to the Quota Server and to the BMA, page 1-45](#)
- [Handling Out-of-Quota Conditions, page 1-45](#)

Charging for Service Duration Billing

Service Duration Billing is charged according to the following rules:

- For TCP sessions, the Last Billable Session Time (LBST) is the timestamp of the end of the session. The end of the session is detected using TCP session termination signaling (RST, FIN/ACK signals) or session idleness. For TCP sessions only, the session idle time is included in the LBST.

Because non-TCP sessions (such as UDP) do not have a Layer 4 session termination mechanism, the LBST for non-TCP sessions is determined based on the time that the last packet was forwarded for the IP session.

- For a service, the First Billable Time (FBT) is the timestamp (in seconds) of the first grant of network access to a session mapped to a duration-based charging prepaid service. Typically, this time is equal to the timestamp of the first Service Authorization Response with a quota other than zero.
- For a service, the Last Billable Time (LBT) is the greatest timestamp (in seconds) of the LBST for all IP sessions mapping to the service for this user. Optionally (and by default), the value for service idle is added to the maximum interval of the LBST when the LBT is calculated. The service idle timeout is added to the duration because the duration calculation already includes the intermediate idle intervals (the ones between IP sessions); this means that the last idle interval is also included.

Calculating Usage for Service Duration Billing

The calculations for Service Duration Billing usage are as follows:

- If the service is ended as a result of service idleness, the calculation for usage is:

$$\text{LBT} - \text{FBT} = \text{Usage}$$

- If the service is ended as a result of an asynchronous event such as user logoff, the calculation for usage is:

$$\text{LBT} - \text{FBT} = \text{Usage}$$

or

$$\text{Asynchronous Event Timestamp} - \text{FBT} = \text{Usage}$$

whichever is smaller.

- If the service runs out of quota, the calculation for usage is:

$$\text{LBT} - \text{FBT} = \text{Usage}$$

or

$$\text{Out Of Quota Timestamp} - \text{FBT} = \text{Usage}$$

whichever is smaller.



Note

If the user runs out of quota, but the user refreshes the quota before the service idles out, the periods (or gaps) of zero quota are not included in the usage calculation.

When a Service Duration Billing service is a member of a billing plan, and an accounting definition is in service and downloaded to a CSG module, you cannot modify the billing basis or meter configuration. You are instructed at the console to configure **no inservice** on the downloaded accounting definitions.

The following examples show how Service Billing Duration usage is calculated in different situations. In these examples:

- SIT is the service idle timer, configured by using the **idle** command in CSG service configuration mode.
- The service idle time is not included in the billing charge.

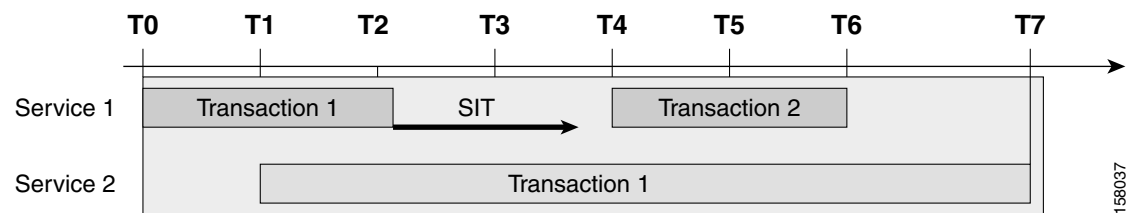
In [Figure 1-1](#), the CSG calculates usage for Service 1 as:

$$(T2 - T0) + (T6 - T4) = \text{Usage}$$

and the usage for Service 2 as:

$$T7 - T1 = \text{Usage}$$

Figure 1-1 Service Duration Billing—Layer 7 Inspection, Two Services, Overlapping Transactions, Example One



158037

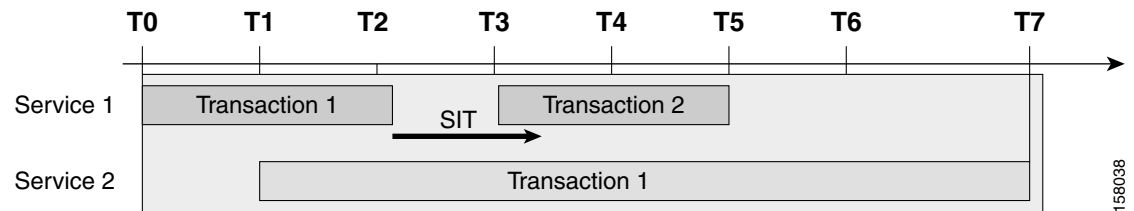
In [Figure 1-2](#), the CSG calculates usage for Service 1 as:

$$T5 - T0 = \text{Usage}$$

and the usage for Service 2 as:

$$T7 - T1 = \text{Usage}$$

Figure 1-2 *Service Duration Billing—Layer 7 Inspection, Two Services, Overlapping Transactions, Example Two*



156038

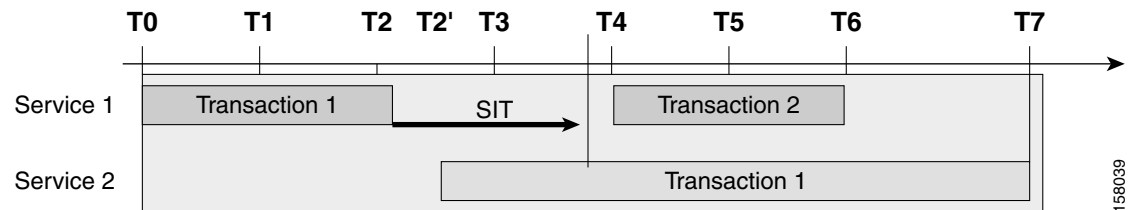
In [Figure 1-3](#), the CSG calculates usage for Service 1 as:

$$(T2 - T0) + (T6 - T4) = \text{Usage}$$

and the usage for Service 2 as:

$$T7 - T2' = \text{Usage}$$

Figure 1-3 *Service Duration Billing—Layer 7 Inspection, Two Services, Non-Overlapping Transactions, Example One*



156039

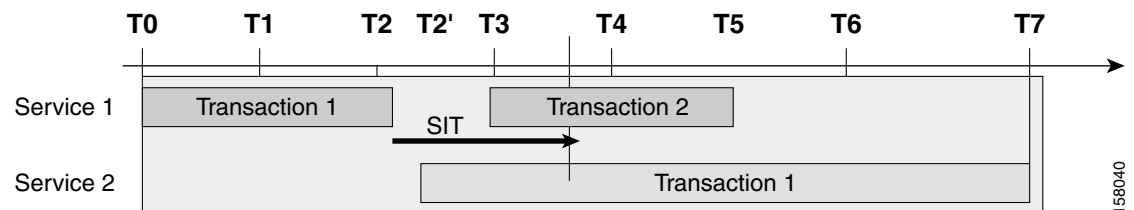
In [Figure 1-4](#), the CSG calculates usage for Service 1 as:

$$T5 - T0 = \text{Usage}$$

and the usage for Service 2 as:

$$T7 - T2' = \text{Usage}$$

Figure 1-4 *Service Duration Billing—Layer 7 Inspection, Two Services, Non-Overlapping Transactions, Example Two*

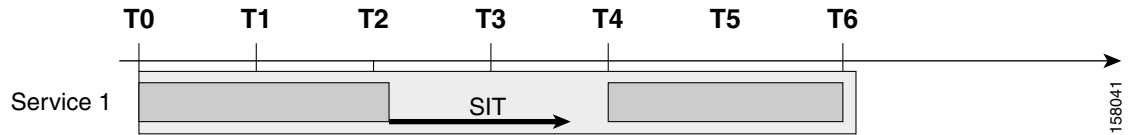


156040

In [Figure 1-5](#), if $T4 - T2 > SIT$, the CSG calculates usage for Service 1 as:

$$(T2 - T0) + (T6 - T4) = \text{Usage}$$

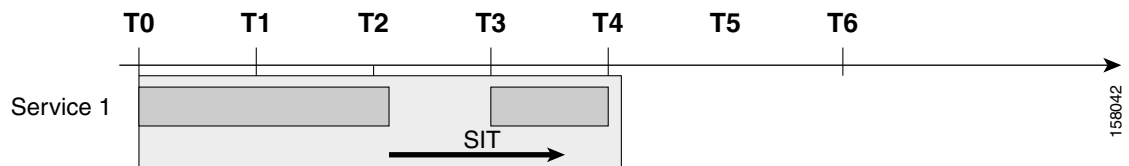
Figure 1-5 Service Duration Billing—Layer 7 Inspection, One Service, Example One



In [Figure 1-6](#), if $T3 - T2 < SIT$, the CSG calculates usage for Service 1 as:

$$T4 - T0 = \text{Usage}$$

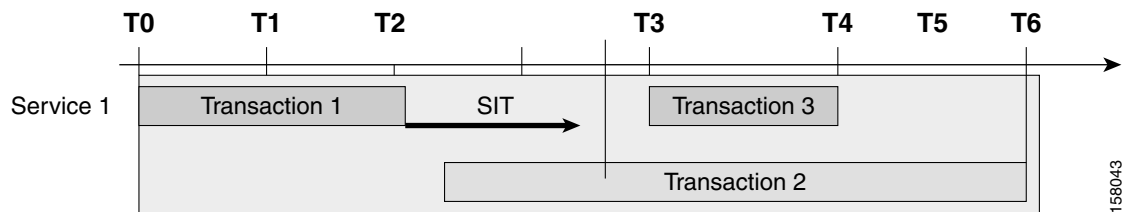
Figure 1-6 Service Duration Billing—Layer 7 Inspection, One Service, Example Two



In [Figure 1-7](#), if the SIT does not time out, the CSG calculates usage for Service 1 as:

$$T6 - T0 = \text{Usage}$$

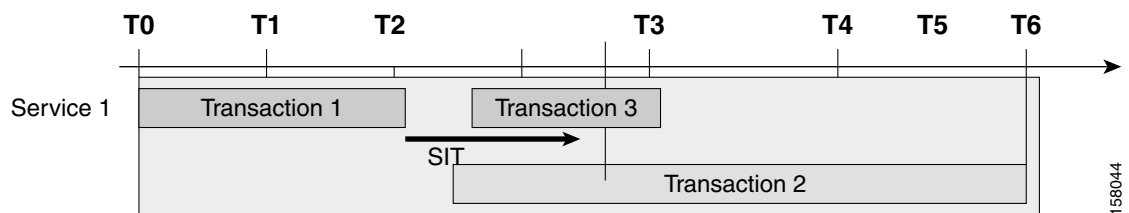
Figure 1-7 Service Duration Billing—Layer 7 Inspection, One Service, Example Three



In [Figure 1-8](#), if the SIT does not time out, the CSG calculates usage for Service 1 as:

$$T6 - T0 = \text{Usage}$$

Figure 1-8 Service Duration Billing—Layer 7 Inspection, One Service, Example Four



Reporting Quadrans to the Quota Server and to the BMA

For Service Duration Billing, quadrans are reported to the quota server and to the BMA in seconds. In messages sent to the BMA on a per-IP-session basis (such as TCP statistics), the prepaid TLVs (Session ID, Service ID, Quadran) are present; the value for quadrans in the Quadran TLV is zero because the duration is based on service, not on individual sessions or on the sum of the durations of the individual sessions.

Handling Out-of-Quota Conditions

When a subscriber runs out of quota, the CSG ends the user sessions that are mapped to the service. The sessions are ended using the same asynchronous session mechanism that is used when a subscriber User Table entry is deleted. The CSG reauthorizes when the remaining time is low (instead of 0) in order to more quickly determine session processing when zero quota is reached.

Connection Duration Billing

Connection Duration Billing enables the CSG to deduct quota based on the time that a user is logged on to the IP network. That differs from Service Duration Billing, which charges on the basis of a service duration. Because the service measures the duration of subscriber access, the service is never idle—it is ended only when the user logs out, or when a Service Stop Request is received from the quota server.

The CSG charges on the basis of the following rules:

- The First Billable Time (FBT) is the timestamp, in seconds, of the first non-zero grant of quota in a Service Authorization Response for the Connection Duration service. A Service Authorization Request is generated when the following conditions are met:
 - A User Table entry is created (typically as a result of a RADIUS Accounting Start message).
 - A Connection Duration service is part of the billing profile for the User Table entry (indicated in a RADIUS Access-Accept message, a RADIUS Accounting Start message, or a Quota Server User Profile Response).

If the user has quota, the FBT is typically the same time as the RADIUS Accounting Start message.

- The Last Billable Time (LBT) is the timestamp, in seconds, when the User Table entry is deleted. During the service lifetime, the CSG updates the LBT when either of the following situations occurs:
 - An IP session starts or ends.
 - The CSG sends a Service Reauthorization Request. This results in an update to the service balance and usage before the Service Reauthorization Request is sent. The CSG uses the following algorithm to calculate the usage:

$$\text{LBT} - \text{FBT} = \text{Usage}$$

or

$$\text{Out Of Quota Timestamp} - \text{FBT} = \text{Usage}$$

whichever is smaller.

Therefore, if the service does not run out of quota, the algorithm is simply as follows:

$$\text{LBT} - \text{FBT} = \text{Usage}$$

If the user runs out of quota, but refreshes the quota before the service times out, the periods of zero quota are not included in the usage calculation. When the user runs out of quota, existing prepaid and postpaid IP sessions for the subscriber are terminated. If the user does not have quota to proceed, no IP sessions are allowed for the user. The CSG provides enforcement only for the policies that have accounting configured.

Connection Duration Billing requires an external quota server. Connection Duration Billing is not supported for internal quota servers, such as a GGSN for postpaid users. Therefore, you cannot use Connection Duration Billing if your quota servers are GGSNs.

To configure Connection Duration Billing, use the **activation** command and the **basis second** command in CSG service configuration mode.

When configuring Service Duration Billing, make sure the content idle timer duration, set using the **idle** command in CSG content configuration mode, does not exceed the service idle timer duration, set using the **idle** command in CSG service configuration mode.

Postpaid Service Tagging

This feature enables the CSG to map postpaid content to a CSG service, and to report the service name in a CSG Service ID TLV in transaction-level CDRs to the BMA. (The CSG Service Session ID TLV is not sent in variable-format records for postpaid service tagging.)

The service must be associated with a billing plan that is configured for postpaid mode. As with prepaid billing plans, the user can be associated with a billing plan via a RADIUS message or a User Profile Response from the quota server. If no quota server is configured, and the billing plan cannot be determined from RADIUS messages, the user is automatically associated with any billing plan configured for postpaid mode. We strongly recommend that you configure only one billing plan for postpaid mode.

Stateful Redundancy and Failover

The CSG supports stateful redundancy for FTP, HTTP, IMAP, POP3, SMTP, TCP, and WAP connections.

Stateful redundancy is the configuration of the active CSG to share information related to billing with its standby CSG in the event of a failure. That is, the session continues to be billed even when the active CSG fails and the standby CSG takes over.

As described in the [“Configuring Fault Tolerance” section on page 4-4](#), the active and standby CSGs use a private VLAN to exchange connection and billing status information. The configurations must be the same on each CSG. We recommend that the quota server, BMA, and user ID database definitions also be the same, although this is not required.

During normal operation, connection and billing state information is sent by the active CSG to the standby CSG, and from the active quota server to the standby quota server. The active CSG and the standby CSGs maintain state information for the configured BMAs. The active CSG keeps the standby CSG informed about which BMAs and quota servers are being used. If the active CSG fails, the standby CSG takes over operation and tries to use the same BMAs or quota servers, if it has connectivity. Otherwise, the standby CSG selects the BMAs or quota servers that have the highest priority.

The active CSG also informs the standby CSG when user IDs are added to or removed from the User Table, and sends the correlators to the standby CSG to ensure consistency when sending billing records for recovered connections to the BMAs. Quota use is also correlated.

If connections are replicated to a standby CSG, and a failover occurs, the CSG does not increment the content counters for traffic flowing through these connections. The CSG does increment the content counters for new connections created after the failover.

The CSG provides full stateful failover for FTP and IMAP sessions.

The CSG provides limited stateful failover for HTTP, POP3, SMTP, and WAP sessions. User information and quota information are maintained on the standby CSG; however, in-process transactions are not. If the active CSG fails, the user transaction is completed on the standby, but no quota is charged for the transaction. Normal billing resumes with the user's next transaction.

The CSG also supports stateful redundancy for TCP connections. That is, the session continues to be billed even when the active CSG fails and the standby CSG takes over.

The CSG does not support stateful redundancy for IP, RTSP, or UDP connections.



Note

Before manually resetting an active CSG, make sure the standby CSG has the complete user and session fault-tolerant (FT) configuration information. In the logs for the active CSG, the following message indicates that the exchange with the standby CSG was successful: "CSG user and session FT dump complete."

"Default" Policy

The CSG matches content on a best-match basis, based on Layer 3 and Layer 4 information. When there is a successful content match, the CSG matches against the policies configured within the content, linearly, on a first-match basis. If no policy within the content matches, the CSG matches against an implicit "default" policy, which matches all traffic. Matching this "default" policy does not generate a CDR, because no accounting policies can be configured for the "default" policy.

For example, assume the following policy and content configuration:

```
ip csg policy PHTTP1
  accounting type http customer-string HTTP-POL1
ip csg policy PHTTP2
  accounting type http customer-string HTTP-POL2
ip csg content HTTP
  policy PHTTP1
  policy PHTTP2
```

The output from the **show module contentServicesGateway 5 content name HTTP detail** command is as follows:

```
HTTP, state = OPERATIONAL, index = 10
destination = 198.133.219.0/24:80, TCP
idle = 3600, replicate = none, vlan = ALL, pending = 30
max parse len = 4000, persist rebalance = TRUE
conns = 0, total conns = 0

policy          total conn  client pkts  server pkts
-----
PHTTP1          0           0            0
PHTTP2          0           0            0
(default)      4760        30056        26534
```

In this example, any TCP traffic that does not match either the PHTTP1 policy or the PHTTP2 policy matches the "default" policy, and this traffic is reflected in the **(default)** row.

**Note**

Even if you have used the **next-hop** command in CSG policy configuration mode to define a next-hop IP address, traffic that matches the “default” policy of a content might not be routed with next-hop.

Prepaid Error Reimbursement

The Prepaid Error Reimbursement feature (also known as CSG quota refund) allows the CSG to automatically refund quota for failed transactions, as defined by the CLI.

- To specify IP, TCP, or wireless application protocol (WAP) flag bit masks and values for CSG quota refund, use the **flags** command in CSG refund configuration mode.
- To specify the range of application return codes for which the CSG refunds quota, use the **retcode** command in CSG refund configuration mode.

The CSG also adds a refund TLV to the statistics records on the BMA interface. The refund TLV is added for transactions that meet a refund condition. The refund amount contains the quota amount to be refunded for the transaction. The refund amount is the same number that is reported in the quadrans TLV. Thus, the full charge for the transaction is always refunded for these protocols.

**Note**

Prepaid Error Reimbursement requires an external quota server. Prepaid Error Reimbursement is not supported for internal quota servers, such as a GGSN for postpaid users. Therefore, you cannot use Prepaid Error Reimbursement if your quota servers are GGSNs.

Prepaid Error Reimbursement is not supported for duration-based services.

Service-level CDRs do not contain quota reimbursement information.

If refund is enabled for a CSG prepaid service, you cannot download more than 0x6FFFFFFF bytes of data in a given transaction.

Support for the Cisco Persistent Storage Device

The CSG supports the Cisco Persistent Storage Device (PSD). The PSD provides persistent storage capabilities to the CSG, and allows the CSG to store data on the PSD’s internal hard drive.

Under normal conditions, the CSG sends content billing records to the mediation partners’ servers. If, for any reason, those servers become unreachable, records are sent to the PSD for safekeeping until contact is reestablished with the Billing Mediation Agent (BMA). Once contact is reestablished, the CSG retrieves the records from the PSD, and forwards them to the BMA.

Storage

Under normal conditions, the PSD provides standby capabilities when necessary—for example, during network outages. The PSD stores the payload from the packet in a queue, and is unaware of the content or format of that data, so that the data can be retrieved exactly as it was sent.

Retrieval

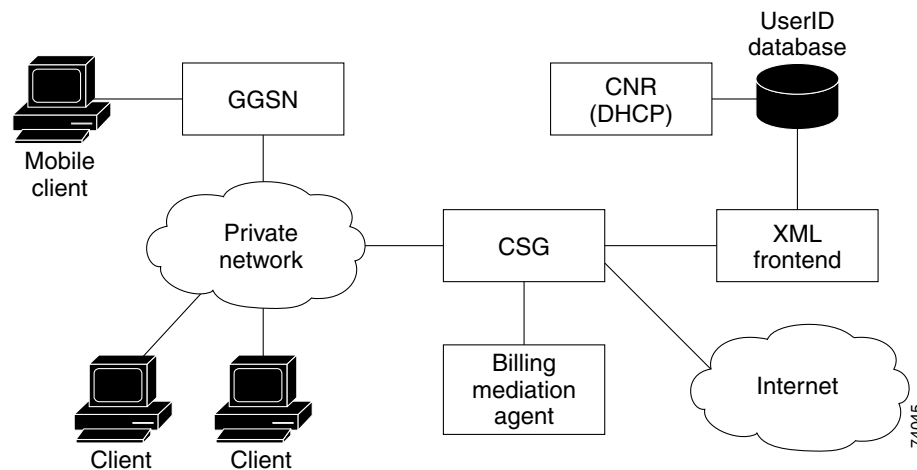
Once the CSG determines that the regular data server is again reachable (in this case, the BMA), it retrieves the stored data from the PSD. The data is returned to the CSG in the same order and form as it was deposited. The CSG is responsible for maintaining order, if necessary, or of mixing retrieved data

with incoming “live” records. Once the CSG acknowledges to the PSD that it has successfully sent the data to the client server (the BMA), the PSD deletes that data. The PSD stores the data until it receives this acknowledgement.

Postpaid Billing

Figure 1-9 shows simple traffic flows between the various components in a simple postpaid CSG environment.

Figure 1-9 Traffic Flow Between Client and Server



Clients send requests that pass through a private network, or through a GGSN, before they reach the Internet.

The CSG monitors data flows and generates accounting records that can be used to bill customers at a content level. The CSG sends the accounting records to a Billing Mediation Agent (BMA), which formats the records as required by the customer’s billing system.

User IDs are obtained from RADIUS Accounting records, or by querying the user database.

BMA Load Sharing

The CSG can support multiple BMAs. This is useful in environments in which the number of billing records sent by the CSG could overwhelm a single BMA.



Note

Multiple BMAs cannot have the same IP address.

The CSG maintains GTP’ sequence numbers for each BMA.

All the billing records for a given user are sent to the same BMA.

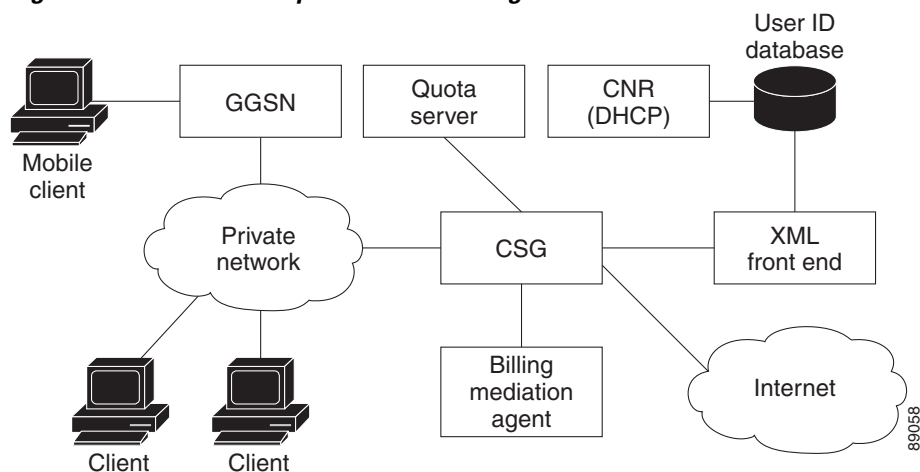
Prepaid Content Billing and Accounting

In addition to postpaid billing, the CSG provides prepaid content billing and accounting. You can configure multiple prepaid billing plans, and subscribers can choose the plan that best meets their needs. Each subscriber can use only one billing plan.

The CSG uses a BMA to interface with a billing server. At the end of each transaction, the CSG sends a billing record to the BMA, indicating the content accessed and the amount deducted. The BMA logs the information in the user's bill.

The CSG uses a quota server to keep track of the quota that is left in the user's account. Each CSG supports one quota server and multiple idle standby quota servers. The CSG allows multiple groups of users on each quota server, with one quota manager for each user. Figure 1-10 illustrates a typical CSG prepaid content billing network.

Figure 1-10 CSG Prepaid Content Billing Network



Quota is provided by the Quota Manager, on request for quota by the CSG. This quota is either for an initial service connection, or for reauthorization when the original or last quota grant is depleted. The Quota Manager can provide a value ranging from 0 to 2,147,483,647 (0x00000000 to 0x7FFFFFFF). This value—called “quadrans”—comes in three forms: **basis byte** for volume-based billing, **basis fixed** for event-based billing, and **basis second** for duration-based billing.

When quota is depleted to zero, the user can no longer access the service.

Quota is held on a per-service basis. Therefore, if a user is connected to more than one service, the CSG stores quota for each service that is open.

After the user finishes a session by closing the bearer session (a RADIUS Accounting Stop message is sent from a GGSN), the service is stopped, and any unused quota is returned to the Quota Manager.

While this prepaid system is in operation, the normal postpaid system runs by sending CDRs to the BMA.

The following flow example describes a basic prepaid flow between the CSG and the Quota Manager:

1. The NAS or GGSN sends a RADIUS Access Request message to the RADIUS Server.
2. On receipt of a RADIUS Access-Accept message from the RADIUS Server, the NAS or GGSN sends a RADIUS Accounting Start message to the RADIUS Server.
3. The CSG creates a user entry and links the user IP address to either the username or Calling Station ID (depending upon the configuration of the CSG).

4. The CSG sends a User Authorization Request to the Quota Manager.
5. The Quota Manager replies to the CSG with a valid billing plan for the user (User Authorization Response).
6. User traffic begins to flow from the NAS or GGSN to the requested server.
7. The CSG sends a Service Authorization Request to the Quota Manager, requesting quota for this connection.
8. The Quota Manager returns a given quota in the Service Authorization Response (if there is quota to give).
9. The user traffic passes the CSG to the service, and the prepaid billing begins.
10. A Service Stop occurs if either the NAS or GGSN sends a RADIUS Accounting Stop message, or if the content and service time out.
11. The Service Stop provides the quota used and returns any remaining quota.

Obtaining User IDs

The CSG uses two methods to obtain user IDs:

- The CSG can use an external user ID database to map IP addresses to user IDs. When the CSG receives a packet with an unknown IP address, and it needs to associate the IP address with a user ID, it queries the database. If the user ID is not available, the CSG generates an accounting record without it.
- The CSG can act as a RADIUS Accounting Server or as a RADIUS proxy for RADIUS Accounting messages. The CSG can examine the accounting messages to determine the user IDs. (The CSG does not support full RADIUS Accounting.)

After identifying a user, the CSG associates the user's IP address with the user ID. If a quota server has been defined, the CSG tries to download the user's profile. The profile indicates whether the user is postpaid or prepaid and indicates the user's billing plan. If the user is a prepaid user, the CSG also downloads the user's quota, and then forwards the user's flows.

Filtering Accounting

Filtering lets you specify the following items:

- Sites to include or exclude for billing information. Specific sites are identified by URL, IP address, protocol, or port parameters.
- A customer string to insert in billing records for the specified site.
- That protocol-specific information is to be generated for billing records to a specified site.

The CSG supports Per-Event Filtering, which permits or denies a transaction as directed by the quota server.

To enable Per-Event Filtering, use the **authorize content** command in CSG service configuration mode.

Intermediate Billing Records

Typically, the CSG sends two billing records for each HTTP session. The CSG sends one record for all non-HTTP sessions, when the sessions end. However, for long-lived sessions, you might want to monitor the progress of the session. To monitor long-lived sessions, you can configure the CSG to send intermediate billing records after a specified number of seconds, or after a specified number of bytes, whichever occurs first.

Intermediate counts are also correlated between the active CSG and the standby CSG.

The CSG supports intermediate billing for FTP, HTTP, IP, TCP, and UDP. The CSG does not support intermediate billing for WAP or e-mail protocols (such as IMAP, POP3, and SMTP). The CSG does not support intermediate billing for RTSP control sessions unless the video/audio traffic is also transported over the control session.

Packet Forwarding

The CSG configurations allow users to specify multiple gateways, one for each VLAN. However, only one default gateway can be in effect at a time. Thus, the second default gateway does not take effect until the first default gateway is unconfigured. We recommend that you specify only one default gateway.

All traffic that passes through the CSG, including the traffic to and from the CSG (GTP' traffic), is routed and forwarded based on the VLAN interface, route, and gateway statements specified by the **module CSG** commands.

For example:

```
module csg 6

vlan 10 server
ip address 10.250.0.1 255.255.0.0
gateway 10.250.1.1

vlan 251 client
ip address 10.251.0.1 255.255.0.0
route 10.200.0.0 255.254.0.0 gateway 10.251.2.11
```

The following logic determines how packets are forwarded:

1. If the destination IP address of the packet is a subnet adjacent to one of the configured VLAN interfaces, the CSG tries to map the destination IP address to a MAC address, using the Address Resolution Protocol (ARP), and forwards the packet.
2. If the destination IP address of the packet is not an adjacent subnet, the CSG forwards the packet based on the routes specified.
3. If there are no matching routes, the CSG forwards the destination IP address of the packet based on the default gateway as defined. If no default gateway is specified, and if the CSG does not have an ARP entry in its ARP cache for the MAC address, the CSG drops the packet.

For packets that originate from the CSG (GTP'), the CSG uses the preceding logic to forward the packet. In the preceding configuration example, if the CSG forwards a packet using the default gateway, the CSG uses the source interface of 10.250.0.1 to forward the packet.

In general, the default gateway is used to specify on the server VLAN to direct client traffic to the Internet. This avoids the need to specify a large number of subnets, because some subnets are not easily identified. The default gateway can be placed in either the client VLAN or server VLAN.

In addition to using route/gateway to forward the traffic, client/server traffic can also be forwarded using next-hop as specified in the billing policy. For example:

```
ip csg policy FORWARD_PKT
  accounting customer-string CLIENT-TRAFFIC
  next-hop 1.1.1.1

ip csg content FORWARD-INTERNET-TRAFFIC
  ip any
  policy FORWARD_PKT
  inservice
```

In this example, if traffic matches this content and policy, the CSG forwards the traffic to the next-hop router that has an IP address of 1.1.1.1.

The CSG supports next-hop packet forwarding for all protocols.

**Note**

Even if you have used the **next-hop** command in CSG policy configuration mode to define a next-hop IP address, traffic that matches the “default” policy of a content might not be routed with next-hop.

Supplemental Usage Reports

You can configure the CSG to report supplemental usage to the quota server when sending a Service Stop, Quota Return, or Service Reauthorization Request message. The supplemental usage data reports the uploaded bytes, downloaded bytes, usage time in seconds, and time stamps for the first and last billable sessions. The data is incremental from the last report. The supplemental usage is included in the Service Stop Notification (0x0011) and in the Quota Return Notification (0x0009).

If a tariff switch timeout occurs during the interval, the CSG sends the tariff switch TLVs along with the supplemental usage TLVs. The supplemental usage TLVs cover the data from the tariff switch time to the end of the interval.

Supplemental usage reporting always reports IP bytes, even if the billing basis is configured for TCP bytes.

To enable supplemental usage reporting, use the **report usage** command in CSG accounting configuration mode.

Enhanced Interoperability with Cisco Service-Aware GGSN

The CSG can couple with a Cisco GGSN to form a service-aware GGSN. When operating in this mode, the CSG gets quota from the GGSN. For more information, see the *Cisco GGSN Release 5.2 Configuration Guide*.

There are no new commands required to enable enhanced interoperability.

Miscellaneous Features

The CSG provides the following miscellaneous features:

- [Support for the CSG MIB, page 1-54](#)
- [Non-HTTP Traffic, page 1-54](#)
- [Fragment Support, page 1-54](#)

- [Report Billing Plan ID to BMA and Quota Server, page 1-54](#)
- [Asynchronous Service Stop, page 1-54](#)
- [Support for Port Number Ranges, page 1-54](#)
- [Learning Client IP Addresses Using Inspection of X-Forwarded-For Headers, page 1-55](#)
- [Negative Quadrans, page 1-55](#)

Support for the CSG MIB

The CSG supports the CISCO-CSG-MIB implemented in the Cisco IOS software.



Note

The CISCO-CSG-MIB agent queries every CSG module on the chassis, so minor delays in responses to SNMP queries are to be expected. To prevent potential problems, you might need to increase the SNMP walk timeout.

Non-HTTP Traffic

For non-HTTP traffic, the CSG records information about data transfer, based on flow information.

Fragment Support

The CSG supports IP fragments for both TCP and non-TCP flows, including fragments that arrive out of order.

Report Billing Plan ID to BMA and Quota Server

The CSG reports the billing plan identifier string both in BMA records and in messages to the quota server.

Asynchronous Service Stop

The Asynchronous Service Stop feature allows the quota server to request the CSG to stop a prepaid service for a defined user and service, and to send a Service Stop.

Support for Port Number Ranges

When you configure content on the CSG, you can define a single port number, or a range of port numbers. This eliminates the need to configure a content for each port.

When defining a range of port numbers, choose a range that is applicable to the associated policies. For example, defining a range of port numbers from 80 to 8080 for **accounting type http** means that the CSG must perform intensive HTTP inspection on many intermediate ports, ports that might not be expected to carry HTTP flows. HTTP inspection of such a high volume of non-HTTP flows can result in excessive processing by the CSG, as well as generating many CDRs that the customer had not planned for.

Learning Client IP Addresses Using Inspection of X-Forwarded-For Headers

If your network is configured with a gateway or proxy placed between the client and the CSG, you can configure the CSG to determine the client's IP address by inspecting the X-Forwarded-For header (for HTTP connections only).

Negative Quadrans

The quota balance in a prepaid service can become a negative value when the user's quota is being depleted, and the billing basis is **byte ip** or **byte tcp**. This can occur because the CSG forwards the entire received packet as long as the service's available quota is greater than 0. If the forwarded packet has more bytes than the quota balance, the balance becomes negative. Note that the CSG might report this negative balance to the quota server as a negative number in the Remaining Quota TLV.

Prerequisites

- The CSG runs with Cisco IOS Release 12.1(12c)E4 or later.
- When you configure redundant CSGs, the standby CSG must use the same software release as the active CSG, or a later software release. If your CSGs act as standbys for each other, they must all use the same software release.
- FTP requires a control TCP connection to well-known server port 21.

Restrictions

- The CSG supports only Internet Protocol version 4 (IPv4). It does not support Internet Protocol version 6 (IPv6).
- The CSG does not support IP packets larger than 1500 bytes.
- The CSG supports up to 4096 content/policy pairs configured under services within a billing plan. (If two or more billing plans contain the same service, the content/policy pairs are counted multiple times.)
- The CSG supports up to 1024 services and up to 4096 services rules.
- The CSG supports up to 4000 content configurations (or virtual server configurations in postpaid); up to 1000 unique IP addresses (virtual IP addresses plus VLAN IP addresses and alias IP addresses); and up to 127 contents for each unique IP address (each content counts as one and each VLAN/alias IP address counts as four).
- Because of limitations on the number of URL match patterns that the CSG can handle, do not define more than 16,000 policies. For more information on URL match patterns, see the description of the [match \(URL map\)](#) command.
- When you enter a new or changed URL match pattern using the [match \(URL map\)](#) command, the CSG console becomes non-responsive while the CSG downloads the entire configuration, which can take a long time. Therefore, we recommend that you configure the URL match pattern during your maintenance window, or during off-peak hours.
- The CSG supports up to 16,000 access control list (ACL) items.
- The CSG supports up to 256 total VLANs (client and server).

- Up to six Cisco CSGs and/or Content Services Modules (CSMs) can be installed in a Catalyst 6500 series switch or Cisco 7600 series router chassis.
- More than one CSG can run in a Catalyst 6000 series switch or Cisco 7600 series router chassis.
- The CSG does not support dual quota (that is, the ability to deduct quota based on multiple criteria at the same time for the same flow).
- For IP Layer 4 and Layer 7 inspection, the CSG volume counters wrap at 0xFFFFFFFF (4294967295 bytes). The volume counters are 32 bits unsigned.
- For Layer 7 inspection for FTP, IMAP, POP3, RTSP, SMTP, and WAP, the CSG does not update the counters until the session ends. For Layer 4 inspection for all protocols, the CSG updates the counters in real time.
- If the CSG memory usage reaches 98% or higher, the CSG stops allocating storage. If this occurs, some CSG billing operations might fail. This 98% threshold is a fixed CSG threshold; it is not affected by the setting of any CSG environment variables, including the `CSG_MEM_MAX_THRESHOLD` variable.
- During multipart or chunked POST processing, the CSG can buffer up to 29,696 packets for all users. Therefore, the maximum theoretical POST size for a single user would be 44.5MB (29,696 packets at 1,500 bytes/packet). However, this theoretical maximum is reduced if other users have active multipart or chunked POSTs in process.
- The CSG reports all times in Coordinated Universal Time (UTC), regardless of the setting of the **clock timezone** or **clock summer-time** command.
- For RTSP, keep the following considerations in mind:
 - RTSP requires a control TCP connection to server port 554.
 - RTSP offers minimal support for TCP-interleaved and HTTP-tunneled transport. Only the first stream URL is reported. For authorized content, the first stream is sent to the quota server. The action must be identical to that sent for the control connection because the stream is interleaved on the control connection and cannot be ended or charged independent of the control connection.
 - RTSP supports only up to two TCP segments. If an RTSP method spans three or more TCP segments, the CSG does not parse the additional segments.
 - RTSP supports multiple transport choices. RTSP clients and servers negotiate the transport choice dynamically before the stream is started.

One transport choice is to interleave the stream with the control channel. In this mode, the CSG cannot map the transport connection to a different policy, and URL mapping cannot be supported.

The other transport choice for RTSP is use of a single HTTP connection. RTSP that is tunneled over HTTP has the same limitation as interleaved RTSP: The stream cannot be mapped to a policy different from the one used by the control connection, as both the stream and the control connection share the same transport.
- The CSG does not support multicast RTSP.
- The CSG does not support return code-based refunding for RTSP, but it does support flag-based refunding for RTSP.
- The CSG does not correlate streams that are described in a container file, such as SMIL.
- The CSG parses only the RTSP control session. When multiple RTSP streams are multiplexed over the same transport channel, the CSG reports cumulative statistics for all the streams.

- If RTSP URL mapping and filtering are used, and if multiple RTSP streams share the same transport channel, the CSG generates a single Content Authorization Request, and the request contains all the URLs carried over that stream. Also, the RTSP stream CDR contains the URLs for all the streams that are multiplexed over the same transport channel.
 - If an RTSP proxy is used, place the CSG on the client side of the proxy. If the CSG is placed on the network side of the proxy, the CSG sees packets originating from the proxy, and the CDRs contain the proxy's IP address, instead of the client's.
 - After a CSG failover, existing RTSP UDP streams continue to operate if a catch-all content rule was defined to pass unknown UDP flows. However, RTSP correlation for those streams ceases, and billing is limited only to the parameters defined for the catch-all content rule. New RTSP connections are processed normally.
 - For RTSP, all policies must use the same access control list (ACL) and the same next-hop IP address.
 - For RTSP, the policy used to determine the next hop address is chosen based solely on ACLs, not URL maps. As a result, you can choose the next hop from one policy for routing and from a different policy for billing.
 - When the CSG is processing high amounts of RTSP traffic, or is clearing a large number of sessions, the message processing levels might exceed the CSG processing capability, forcing the CSG to stop forwarding traffic and reboot.
- For FT CSG pairs:
 - Each FT CSG pair must use a different FT VLAN.
 - If you have pairs of CSG cards and pairs of CSM cards in your network, each pair must use a different FT VLAN. Do not configure a CSG pair and a CSM pair to use the same FT VLAN.
 - The CSG does support trunked FT VLANs, but each pair of CSGs must use a unique FT VLAN and a unique group ID. In addition, make sure that the number of high availability messages between all pairs of CSGs on the trunk does not overwhelm the CSG card.
 - A single CSG environment does not require a route in the CSM that points to the CSG RADIUS virtual IP address. However, in a fault-tolerant setup or a multiple-CSG setup, a route to the CSG RADIUS virtual IP address is required. This route must point to the alias IP of the appropriate CSG.
 - The CSG drops traffic from an unknown source MAC address on the client-side VLAN or the server-side VLAN.
 - The IP address in the content configuration cannot be in the same subnet as the IP address of the client VLANs.
 - Advice of Charge (AoC) via content authorization and URL-redirect is supported for only HTTP and WAP/WSP.
 - When a CSG prepaid service is configured for Advice of Charge (AoC), the weighting value for charging the content is not determined until the CSG processes the Content Authorization Response. For SMTP billing (**accounting type smtp**), the CSG does not send the Content Authorization Request until it processes the SMTP DATA command. If the CSG does not process the SMTP DATA command for a session, then the CSG does not charge the session for volume and event billing.
 - The CSG does not support multiple protocols under a single service definition. Do not configure a CSG service with more than one accounting protocol type.
 - Service verification is supported only for HTTP and WAP.
 - For WAP, keep the following considerations in mind:

- All policies must use the same access control list (ACL) and the same next-hop IP address.
- For WAP 1.x, the policy used to determine the next hop address is chosen based solely on ACLs, not URL maps. As a result, you can choose the next hop from one policy for routing and from a different policy for billing.
- The CSG supports only URL maps for WAP; header maps are not supported. You cannot use the CLI to configure header maps for WAP services. Policies defined as **accounting type wap** can accept only URL map definitions. For WAP 1.x, URL maps take precedence over access lists.
- The CSG does not support the CLOSING or TIME-WAIT states for TCP connections. After the end-points exchange FIN_ACK messages, the connection is terminated immediately, and the CSG does not process any out-of-order packets for the connection.
- For RADIUS:
 - The RADIUS Accounting Start message which specifies the NAS IP to which to send the PoD message must be received on an IP address specified by the **radius proxy** or **radius endpoint** command configured in module CSG configuration mode.
 - Retransmitted RADIUS Accounting Stop messages might cause problems when associating traffic with a subscriber. To avoid any problems, do not configure your RADIUS server to reuse an IP address immediately after it is released by a user.
 - For Enhanced Radius Proxy, in order for the CSG to act on the optional Quota Server TLV in a RADIUS Accounting Start message, the referenced quota server must be manually configured prior to receiving the RADIUS Accounting Start message that contains the TLV.
- For CSG type=HTTP parsing, the CSG imposes the following restrictions:
 - The HTTP method must be initiated by the same endpoint that initiated the TCP connection (by the same side that sent the TCP SYN); the client request transfers no data.
 - The maximum HTTP transaction volume is 268435455 bytes. If this length is exceeded, the CSG invokes Layer 4 billing for the remainder of the connection.
 - HTTP request parsing is limited to 64,000 bytes. Any headers beyond this limit are not recognized and are not used in matching URL or header maps.
 - HTTP and HTTPS cannot share the same port. However, SSL can be tunneled over HTTP using the Connect method.
 - There are two types of maps for HTTP: URLs and headers.
 - If policy type=http, the only TCP option that is passed through the CSG is the maximum segment size (MSS). The other options are dropped. This includes Wireless Profiled TCP Options (example: SACK) that are used with WAP 2.0 implementations.
 - With RFC 2818, an HTTP session can become encrypted via the UPGRADE method. If Layer 7 billing is defined for the HTTP port, the session might time out when the UPGRADE occurs, because the CSG code cannot parse the encrypted data after TLS negotiation.
 - For an HTTP transaction, if any quota is granted, the CSG always sends the following packets, even if there is insufficient quota:
 - The first request (GET, POST, any other method) from the client—headers plus the part of the message that arrives before quota is granted.
 - The headers plus one packet of the response from the server.
 - For HTTP Layer 7 inspection, the CSG supports up to 65,535 concurrent HTTP TCP connections.

- Some HTTP Layer 7 methods and content types cause the CSG to invoke Layer 4 processing for the remainder of the TCP connection. For details, see the HTTP compliance exceptions in the “[Layer 7 Inspection \(accounting type=specific protocol\)](#)” section on page D-1.
- When the CSG is processing a large number of requests for new pipelined HTTP connections, or is clearing a large number of sessions, the message processing levels might exceed the CSG processing capability, forcing the CSG to stop forwarding traffic and reboot.
- The CSG hybrid mode (that is, the CSG running in hybrid mode, with CatOS on the Supervisor Engine and Cisco IOS on the Multilayer Switch Feature Card [MSFC])
 - Runs with Cisco IOS Release 12.1(13)E or later and CatOS 7.6.1 or later.
 - Supports only the CSG Release 2.2(3)C2(1) command-line interface (CLI), not the CSG Release 3.1(1)C3(1) CLI.
 - Does not support the **hw-module module slot reset** command. To reset the CSG hybrid mode, enter the **set module power [up | down] slot** command at the CatOS console.
- When replacing an adjacent device but retaining the same IP address (so that there is a different MAC address but the same IP address as before), you must either enter the **clear module csm slot connections** command in privileged EXEC mode or you must recycle the CSG.



Note The **clear module csm slot connections** command clears all connections for the CSG specified by the slot number; you cannot use this command to clear selected connections.

- Services that are configured with the **basis second connect** command (that is, for Connection Duration Billing) are subject to the following restrictions:
 - Service verification is not supported for Connection Duration services.
 - Advice of Charge (AoC) is not supported for Connection Duration services.
 - If redirect is to be performed when the Connection Duration service runs out of quota, the URL location to which the CSG redirects must map to a policy that does not have accounting configured. This is because all IP sessions mapped to policies with accounting configured (postpaid or prepaid) are dropped when the Connection Duration service has no quota.
- For Service Duration Billing:
 - Content idle is not included for non-TCP connections. Therefore, the idle timeout for non-TCP content configurations is restricted to be less than the service idle timeout of any service that includes the non-TCP content configuration, and that is configured for **basis second**.
 - The CSG does not allow you to specify weights for Service Duration Billing.
- If the CSG does not have an ARP entry in its ARP cache for the MAC address of a device or firewall, it drops any packets that it receives from that device or firewall.
- If you configure content with a network mask of 255.255.255.255 or /32, then a virtual server is created and the CSG MAC address is entered as the host address in the CSG ARP cache. This means that you cannot have hosts directly connected to the CSG, coupled with content with a network mask of 255.255.255.255 or /32 that matches those hosts.
- The CSG does not decrement the time to live (TTL) of an IP packet.
- For Interface Awareness:
 - To enable the CSG to route network-initiated connections correctly, the AAA or NAS must send the `downlink_nextthop` VSA.
 - All downlink next-hop addresses must be configured as gateways in the CSG’s routing table.

- To avoid routing ambiguity on the uplink (or server VLAN) side of the CSG, next hops must override the CSG's routing table.
- Table IDs and names are not supported or reported in fixed-format TLVs.
- If a content configuration is required on multiple VLANs, you must define the content multiple times, once for each of the VLANs on which it is required. Contents cannot be shared across tables.
- VLAN-specific content configurations must handle all traffic from users arriving on a VLAN marked with a table name. The CSG uses the VLAN table name to locate user entries. Therefore, if you want to apply the same contents to multiple tables, you must redefine all of your contents.
- Configurations with overlapping IP address requirements must use CSG RADIUS proxy or RADIUS endpoint to populate the User Table. RADIUS monitor, user database, and old-style RADIUS proxy and endpoint configuration do not support table names or overlapping IP addresses.
- To identify a user within a CSG table, the quota server-initiated messages must contain the Extended User Index TLV to identify or trigger action for a user within a CSG table.
- The CSG supports overlapping subscriber IP addresses, but does not support overlap in interface or gateway configuration. The entities that assign IP addresses to subscribers within the VRF must be aware of all of the restricted addresses which belong to the CSG and to adjacent network devices.
- Network-initiated connections are routed via the default routing table, unless the RADIUS VSA containing the downlink gateway is present in the initial RADIUS flows.
- For Quota Push, the CSG rejects the Quota Push message if the "Replace current balance" flag is not set in the Granted Quadrans TLV.
- For Differentiated Services Code Point (DSCP) tagging:
 - For HTTP, the CSG does not preserve the DSCP tagging for any packets that it generates (such as SYNs sent to a server or SYNACK/ACKs sent to a client).
 - For all other protocols, the CSG is transparent to DSCP tagging. When the CSG forwards a packet, it forwards the DSCP value without modification.
- For Tariff Switch:
 - If CSG refunding is configured for a prepaid service, the tariff switch usage might not include usage on existing IP sessions at the tariff switch time. This is because the usage cannot be charged until the session ends and the refund conditions are evaluated.
 - If a transaction spans multiple tariff switches, but the CSG does not support intermediate records for that protocol, the CSG reports only the most recent tariff switch information.
 - If a transaction is configured for BMA reporting using a fixed-format record, the tariff switch usage information is not reported in the record.
- For obscuring the IP address in X-Forwarded-For headers:
 - The CSG does not obscure the IP address in fragmented request packets that have X-Forwarded-For headers, because the CSG does not reassemble the fragments and therefore cannot modify the packets.
 - The CSG does not obscure the X-Forwarded-For header for traffic that is downgraded from Layer 7 inspection to Layer 4 inspection.

- If the active CSG fails over to the standby CSG, the standby CSG does not obscure the IP address in X-Forwarded-For header for existing HTTP sessions. However, the standby CSG does obscure the IP address in X-Forwarded-For headers for new HTTP sessions.
- If the client sends more than one GET request with X-Forwarded-For headers, and the content host fails to send a TCP acknowledgement within five seconds, the CSG resets the client side connection.
- In a Home Agent (HA) configuration in which the active CSG is running at 3.1(3)C6(2) or later and the standby CSG is running at 3.1(3)C5(5) or 3.1(3)C5(4), time-based billing might result in over-charging.
- With the **replicate connection tcp** command configured, when a connection is established or terminated, the active CSG sends a dummy SYN or RST, respectively, to the fault-tolerant VLAN. This is normal operation. The extra packets are not billed and the destination MAC address is unknown, so the packets do not reach the server. The destination MAC address for the dummy SYN or RST frame is structured as follows:

0x03:xx:yy:00:zz:zz

where:

- **0x03:xx:yy** is the Cisco Organizational Unique Identifier (OUI).
- **zz** is the VLAN of the SYN that initiated the connection.
- If all quota servers fail, the CSG begins storing Service Stop Requests, in order to forward them when a quota server comes back online. In an Intelligent Packet Solution (IPS) environment, when the GGSN quota server comes back online and the CSG forwards the stored Service Stop Requests, if the user's packet data protocol (PDP) context is no longer active or is not known to the Diameter-Closed Loop Charging Interface (D-CLCI) backend, the GGSN quota server might respond to the request with GTP reject code 201.

