



Additional Information about Billing Support

The following sections describe billing and its many aspects. It is critical to understand all SBC billing features and capabilities before performing billing configurations for the SBC.

- [Standalone Billing Systems](#)
- [Integrated Billing Systems](#)
- [Event Message Transmission](#)
- [Call Detail Records](#)
- [Administration and Configuration](#)
- [Logging and Alarms](#)
- [Fault Tolerance](#)
- [Security](#)

Standalone Billing Systems

Standalone billing and billing caching is supported in both a unified model SBC, or a standalone SBE. A standalone billing system comprises the following modes of operation and events:

- When a call starts, SBC begins recording billable events for that call.
- At the end of the call, SBC stops recording, and collates the events into a single call detail record (CDR).
- The CDR is stored on disk. Available disk space constrains the number of CDRs that can be stored. For example: 24 hours of records could take up approximately 10 GB of disk space.

In the event that disk space becomes unavailable, an alarm log in the form of a Simple Network Management Protocol (SNMP) trap is generated, requesting that an administrator free up disk space by removing CDRs. No further CDRs are logged until more disk space is made available, but the system continues to accept calls.

A set of thresholds are configured on SBC, which defines a progression of alarms triggered by increasing file size; this enables an administrator to free up disk space before it runs out.

- The format of the CDRs is in extensible markup language (XML) format, which can be parsed into the format required by the target billing platform.

The CDR format most often used by softswitch vendors to generate CDRs is the Bellcore AMA Format, described in *Billing Automatic Message Accounting Format (BAF) Generic Requirements* (BAF-GR-1100-CORE). Unfortunately for the next-generation of Voice over IP (VoIP) applications,

such as SBC, the BAF format is too telephony specific, and does not contain sufficient provision for IP-centric logging information. (For example, it does not allow for logging of Session Description Protocol (SDP) or RTCP statistics). In addition, the format is not extensible, so it is not possible to define extensions to contain these fields.

An XML format is a desirable alternative, because XML is a flexible, standardized methodology and is commonly used where data must be translated between different platforms (such as between an SBC and a billing server). See the [“End-to-End SBC Configuration Example on a Cisco XR 12000 Series Router”](#) module for more information.

Integrated Billing Systems

Integrated billing is achieved through the PacketCable Event Messages architecture (see the *PacketCable 1.5 Event Messages Specification*; PKT-SP-EM1.5-I01-050128) as exemplified in [Figure 18](#) where the SBC is integrated into this architecture. As shown, the billing server and softswitch both support PacketCable Event Messages.

ISP-A shows SBC operating in a unified model where the billing system is being deployed as a distributed billing system consisting of three billing servers. The SBC can be configured to send to these servers in a range of ways, such as to all three simultaneously, or to use one primary and two backups.

In the unified model, the system operates as follows:

- The SBC produces event messages (EMs). These event messages are billable or other interesting events, such as call start, call end, and media-type changes.
- The SBC (and other elements of the system), which produces EMs, sends them in real time (or batched up for network efficiency) using the RADIUS protocol to the billing server.
- The billing server collates EMs into call detail records.
- In the event that a billing server(s) is unavailable, the EM is marked as being unsent and is stored for up to 24 hours. (The EMs are stored on the Cisco XR 12000 Series Router hard disk depending on the free space available.)
- An alarm log is generated, and the EM is resent to the RADIUS servers by manual CLI commands when the RADIUS servers are back online.

ISP-B shows SBC operating in a distributed model where the billing system is being deployed using a single billing server and a softswitch.

In the distributed model, the system operates as follows:

- Only the SBE communicates with the billing server. That is, no event messages are generated by the DBE. All media-specific information (for example: gate request information and media statistics) is sent by the DBE to the SBE which then generates event messages as required to send to the billing servers.
- The billing server collates billing information both from the SBE and the softswitch to provide the ISP with a single billing point. The softswitch only interface to the billing service is one of the ways service providers could use to get billing information. It is outside the scope of SBC billing.
- In the event that a billing server(s) is unavailable, the EM is marked as being unsent and is stored for up to 24 hours. An alarm log is generated, and the EM will be resent by the billing component when the RADIUS server is back online.



Note

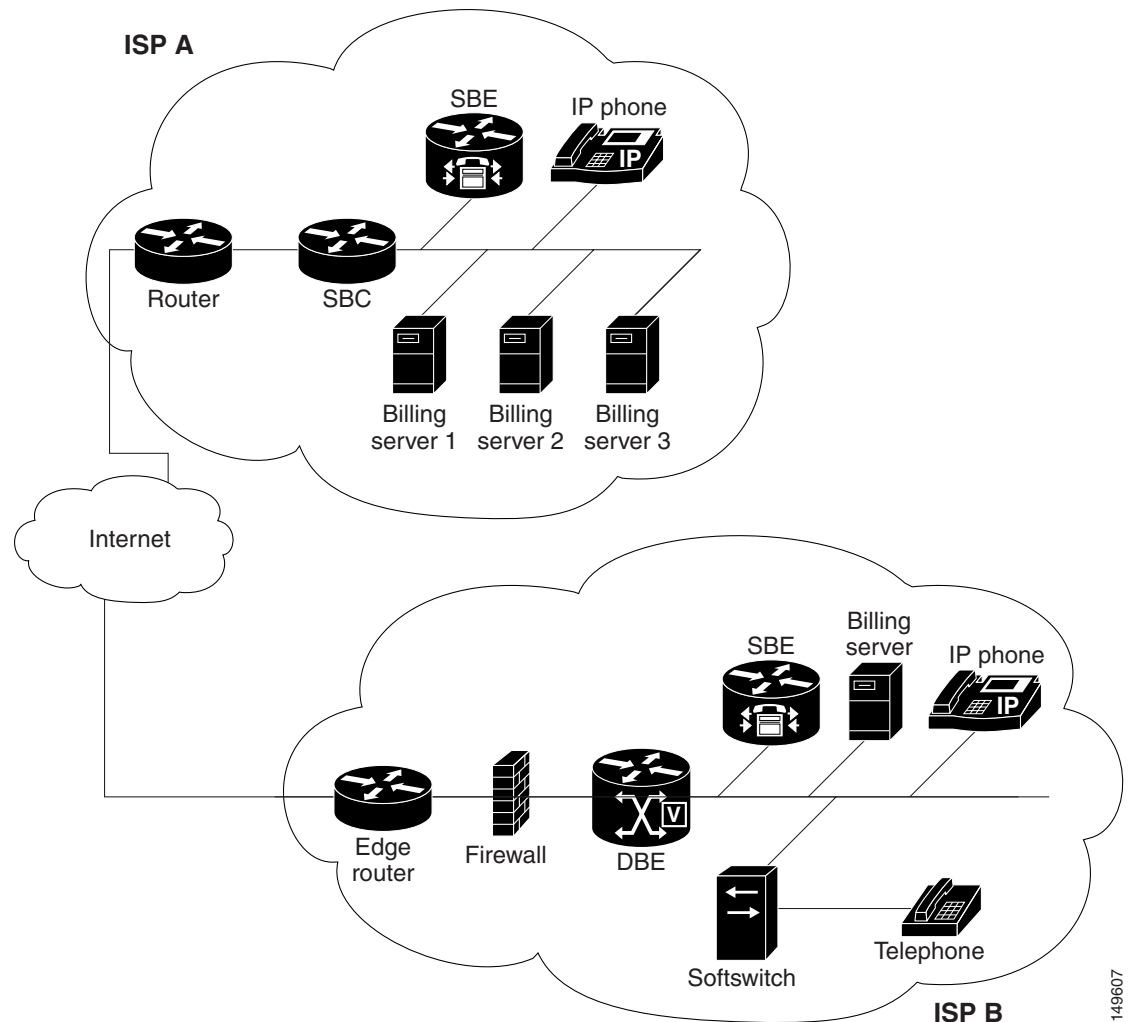
Standalone billing and billing caching is supported in both a unified model SBC, or a standalone SBE.



Note

The *PacketCable 1.5 Event Messages Specification* discusses sending the identifying information (the BCID and FEID) on the outgoing INVITE and responding SDP so that correlation can be done between the two sets of billing data. SBC does not support this mechanism for intra-domain or inter-domain transmission. The billing server must perform the correlation using an alternative method (for example, using the telephone numbers dialed and the time of the call).

Figure 18 Integrated Billing Deployment



Event Message Transmission

The generated event messages, as described in the “[Event Messages Set Overview](#)” section are sent using the RADIUS protocol to a preconfigured set of billing servers. Before getting into the actual detail of the event messages, review the following event message transmission considerations described in the following sections:

- [Multiple Server Support](#)

- [Event Message Batching](#)
- [Resending Event Messages](#)

Multiple Server Support

Billing servers are configured at start-up, in sets:

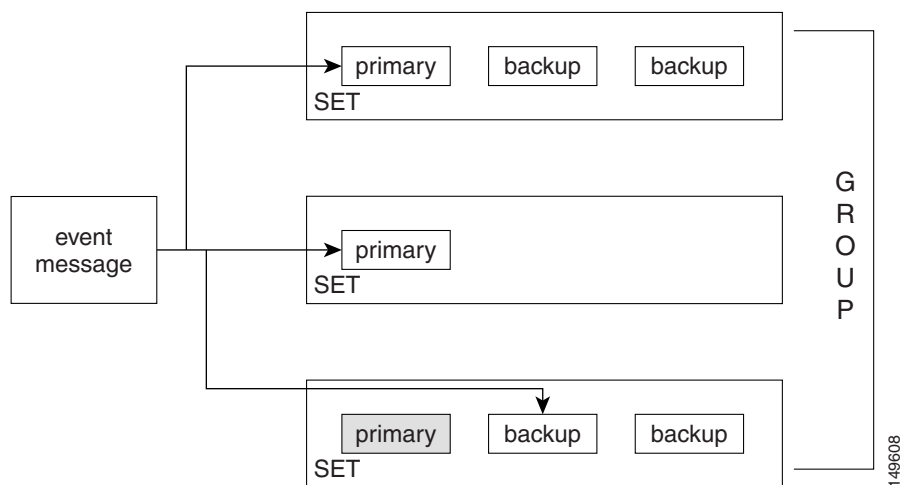
- Each SET contains a list of one or more billing servers, consisting of a single primary server and an ordered list of zero or more backup servers.
- The SBE can be configured with one or more sets of billing servers.

Each event message is sent to the entire collection of sets, but to only one machine within each set.

- For each set, the SBE sends the event message to the primary server within the set.
- If the primary server is unavailable, then the message is sent to the first backup server (if present). If the first backup server is also unavailable, the message is sent to the second backup etc. until either a machine accepts the message or all the servers in the set have been tried.
- If there are no machines in a set accepting messages, then the entire set is marked as unavailable, and messages are cached for that set (to be resent later).

Multiple server support is illustrated in [Figure 19](#).

Figure 19 Multiple Server Support



Event Message Batching

Because of the inefficiency of the RADIUS protocol, the SBE collates event messages into batches and sends them using a single RADIUS message to alleviate the burden on the transport mechanism.

Batching is possible only on a per-set basis. The batch size is not configurable, but is determined by the load on the billing component.

It is not possible to disable batching.

Resending Event Messages

In the event that a set of billing servers does not respond, the following procedure results in the missed events being replayed to the server:

- No billing server within a set accepts an event message.
- The SBE marks the set as unavailable and begins storing event messages for the failed set (beginning with the event messages that have just failed). Messages are stored for up to 24 hours.
- The SBE produces an alarm log. This alarm contains information about which set of servers are not responding (see the “[Logging and Alarms](#)” section).
- The administrator fixes the problem, and uses the command-line interface (CLI) or an SNMP client to trigger a resend attempt. (If the problem cannot be corrected, the cache can also be cleared using this mechanism.) The event messages are replayed from the cache to the (now repaired) set of billing servers and are deleted from the cache when successfully resent.



Note

The messages are not resent immediately, but only when the SBE has resources available that are not required by higher-priority processes (such as real-time billing to non-failed sets). For example, if the administrator were to trigger the resend at 6:00 p.m. local time (often a busy period), the failed messages may not be resent until the network traffic is lighter at (say) 1:00 p.m.

Event Messages Set Overview

This section specifies the set of event messages supported by SBC:

- [Call-Specific Messages](#)
- [Out-of-Band Messages](#)
- [Unsupported Messages](#)

Call-Specific Messages

The messages listed in [Table 18](#) are supported call event messages.

Table 18 **Supported Event Message Sets**

Event Message	Notes
Signaling_Start	Sent when signaling has begun (inbound) and when it is about to begin (outbound); for example, received INVITE on inbound and about to send INVITE on outbound for a SIP endpoint
QoS_Reserve	Sent when there is reserved QoS in the DBE. Sent for the inbound leg when the inbound QoS is reserved, and for the outbound leg when we reserve the outbound QoS is reserved.
Call_Answer	Indicates that the terminating party has answered and that media has started. This message is sent for both legs at the same time.
QoS_Commit	Sent when QoS is committed by the DBE. This message is sent for both legs at the same time.
Call_Disconnect	The call has been terminated; media has ceased flowing. Sent for both legs at the same time.

Table 18 Supported Event Message Sets (continued)

QoS_Release	The QoS has been released by the DBE. Sent for both legs at the same time.
Signaling_Stop	All signaling is complete for each party in the call. (The event is generated once for each party, when the last signaling message has been sent.)
Media_Statistics	Media statistics for the call as reported by the DBE. This is sent for each leg when the media is released.
Media_Alive	Indicates that a long-duration call is still active. This is sent for each leg of the call, at a preconfigured time of day, every 24 hours.

Out-of-Band Messages

The event messages listed in [Table 19](#) are non-call-related, out-of-band event messages.

Table 19 Out-of-Band Event Message Sets

Event Message	Notes
Time_Change	Sent when changes of more than 200 ms occur in the time; also sent for daylight savings changes, and so on.

Unsupported Messages

The event messages listed in [Table 20](#) are not supported.

Table 20 Unsupported Event Messages

Event Message	Notes	Why Not Supported?
Database_Query	Sent when querying external databases about toll-free carriers, LNP routing, and so on.	SBC does not support database queries.
Service_Instance	Indicates an instance of a service.	SBC does not support services. (Services are more applicable to softswitches and application servers.)
Service_Activation	Indicates service activation.	
Service_Deactivation	Indicates service deactivation.	
Interconnect_Start	Sent when interconnecting to PSTN.	SBC does not interface directly to the public switch telephony network (PSTN).
Interconnect_Stop	Sent when terminating a connection to PSTN.	
Conference_Party_Change	Indicates a party state change in a multi-party call.	SBC does not support multi-party calls.

Call Detail Records

The call detail record (CDR) mode of operation does not use a network protocol to generate discrete messages in real time (unlike the integrated mode); instead, at the end of each call, SBC generates a call detail record in a proprietary XML format.

This CDR is appended to a file on disk, which is periodically flipped so that it may be retrieved using FTP (or some other file access mode supported by the underpinning operating system, such as SCP, NTP, and so on.)

This mode of operation is analogous to that provided by traditional class 5 switches. There are three types of CDR logged:

- *Basic CDR*—This record is the most common type of record and logs a completed call.
- *Partial CDR*—This record occurs only when the system has attempted to recover from a catastrophic failure and is logging as much information about an ending call as possible.
- *Long duration record (LDR)*—This record occurs every 24 hours for each call that has been in progress for more than 24 hours. The time of day at which the LDRs are generated is configurable.

In addition to these three, an audit log is also made every hour, summarizing the billing activity over the last hour. The format of each record type, the precise XML DTD definition, as well as file access and media requirements, are described in the following sections:

- [CDR Format](#)
- [LDR Format](#)
- [Partial Call Format](#)
- [Audit Log Format](#)
- [XML DTD](#)
- [File Access](#)
- [CDR Media Requirements](#)

CDR Format

The format of the records is described in the following sections:

- [XML Record Analysis](#)
- [XML Elements](#)



Note

The reason a proprietary format has been adopted is that there is as yet no open-standard VoIP-centric format for CDRs.

XML Record Analysis

An XML record for a short, 90-second call record follows.



Note

For clarity, the example that follows contains extra whitespace; in a production environment, all extraneous whitespace is removed to reduce record size.

```
<?xml version="1.0" ?>

<recordfile sbe="192.49.2.2">
  <!-- BEGIN CALL RECORD -->
  <call starttime="1110916754000"
    endtime="1110916844000"
    duration="90000" bcid="01234567890">
```

```

<party type="orig" phone="02083661177"/>
<party type="term" phone="02083677012"/>

<adjacency type="orig" name="csi_enfield" account="csi" vpn="csivpn"/>
<adjacency type="term" name="softswitch1" account="internal"/>

<connect time="1110916754150"/>
<disconnect time="1110916843790" reason="1"/>

<QoS reservetime="1110916754100"
  committime="1110916754120"
  releasetime="1110916843800">
  <gate>
    <flowinfo>
      <local address="67.12.43.123" port="46512"/>
      <remote address="67.84.141.2" port="44684"/>
      <sd>
        m=audio 0 RTP/AVP 0
        a=rtpmap:0 PCMU/8000
        a=ptime:20
      </sd>
      <RTCPstats>
        PS=1245, OS=62345, PR=1362, OR=68095, PD=0, OD=0, PL=0, JI=0, LA=48,
        PC/RPS=1362, PC/ROS=68095, PC/RPR=1245, PC/RPL=0, PC/RJI=0, PC/RLA=33
      </RTCPstats>
    </flowinfo>
    <flowinfo>
      <local address="172.19.8.45" port="48152"/>
      <remote address="172.23.65.41" port="47132"/>
      <sd>
        m=audio 0 RTP/AVP 0
        a=rtpmap:0 PCMU/8000
        a=ptime:20
      </sd>
      <RTCPstats>
        PS=1362, OS=68095, PR=1245, OR=62345, PD=0, OD=0, PL=0, JI=0, LA=44,
        PC/RPS=1245, PC/ROS=62345, PC/RPR=1362, PC/RPL=0, PC/RJI=0, PC/RLA=31
      </RTCPstats>
    </flowinfo>
  </gate>
</QoS>
</call>
<!-- END CALL RECORD -->
</recordfile>

```

The XML file has the following hierarchy:

- A root document element, recordfile. This element describes the SBE recording the CDRs and contains a list of records.
- The root document element contains 0 or more call elements, each representing a single CDR. In turn, the call element contains the following:
 - Two party elements, indicating the originating and terminating endpoints. There must be two and only two of these—one of each type.
 - Two adjacency elements, indicating the incoming adjacency over which the call was received and the outgoing adjacency over which the call was routed. The account name and (where appropriate) VPN ID are also given. There must be two and only two of these—one of each type.
 - An optional connect element, indicating that the call was connected.

- An optional disconnect element, indicating when and why the call was disconnected. This element is present only if the connect element is also present.
- One or more QoS elements, indicating that QoS was reserved, committed and released in the media plane. Multiple QoS elements indicate that the QoS changed during the call.

Each QoS element contains a number of gate elements (one for each gate required for the call, where a gate is defined to be an instance of a single media stream traversing a media gateway). Each gate element contains two, and only two, flowinfo elements, which describe the flows that terminate on either side of the gate. Each flowinfo element contains the following:

- The local address and port from which packets on this side of the gate are sent and on which packets on this side of the gate are received.
- The remote address and port to which packets on this side of the gate are sent and from which packets on this side of the gate are received.
- A session descriptor, which describes the codec parameters of the flow on this side of the gate. This includes the relevant m= line from the SDP for the flow (excluding the port number) and all associated a= lines. A session descriptor is given separately for each side of the gate, because the gate may transcode the flow.
- A Real-Time Control Protocol (RTCP) statistics element. [Table 21](#) lists the statistics that are reported.

**Note**

All the PC/* values are deduced by inspecting RTCP packets received from the remote endpoint. If no RTCP packets are received (because the remote endpoint does not send them, or because an intermediate device is dropping them), then all these values still are present in the CDR, but are set to zero. All other values are measured by the media gateway, and so are not dependent on RTCP.

Table 21 *RTCP Statistics Element Abbreviations*

Abbreviation	Definition
PS	Number of RTP packets sent to the remote endpoint on this flow.
OS	Number of octets in RTP packets sent to the remote endpoint on this flow. Includes IP/UDP/RTP headers, but not Layer 2 headers.
PR	Number of RTP packets received from the remote endpoint on this flow.
OR	Number of octets in RTP packets received from the remote endpoint on this flow. Includes IP/UDP/RTP headers, but not Layer 2 headers.
PD	Number of RTP packets received from the remote endpoint and discarded locally.
OD	Number of octets in RTP packets received from the remote endpoint and discarded locally. Includes IP/UDP/RTP headers, but not Layer 2 headers.
PL	Number of packets that were not received on the connection, as deduced from gaps in the sequence number.
JI	Average jitter of packets received from the remote endpoint on this flow. Units are milliseconds.
LA	Average latency as measured by the MG, in milliseconds, expressed as an integer number.

Table 21 *RTCP Statistics Element Abbreviations (continued)*

Abbreviation	Definition
PC/RPS	Number of RTP packets sent by the remote endpoint to this MG on this flow. Comparing this with the local number of RTP packets received from the remote endpoint gives an indication of how many incoming packets were dropped on this leg of the call.
PC/ROS	Number of octets in RTP packets sent by the remote endpoint to this MG on this flow. This octet count includes RTP payload bytes, but not IP/UDP/RTP headers.
PC/RPR	Number of RTP packets received by the remote endpoint from this MG on this flow. Comparing this with the local number of RTP packets sent to the remote endpoint gives an indication of how many outgoing packets were dropped on this leg of the call.
PC/RPL	Number of RTP packets reported as lost by the remote endpoint on this flow.
PC/RJI	Average jitter of packets being received on this flow as reported by the remote endpoint. Units are milliseconds.
PC/RLA	Half of the round-trip delay between this MG and the remote endpoint on this flow. Units are milliseconds. The round-trip delay is the time taken for a packet sent from this MG to reach the remote endpoint and be returned to the MG. Summing the ep_round_trip_delay field for this flow with the value from its partner flow gives an indication of the end-to-end, round-trip delay for this flow pair.

XML Elements

Table 22 lists the XML elements and their attributes.

Table 22 *SBC Billing CDR XML Format Elements*

Element	Attribute	Optional	Description
recordfile	sbe	N	IP address of the recording SBE.
call	starttime	N	Time the call began (that is, when signaling began).
	endtime	N	Time the call ended (that is, when signaling ended and resources were released).
	duration	N	Length of the call (in ms).
	bcid	N	Unique (to this SBE instance) identifier for this call record.
party	type	N	One of: <ul style="list-style-type: none"> orig (indicates that this party is the originating endpoint of the call) term (indicates that this party is the terminating endpoint of the call)
	phone	N	Phone number of the party.

Table 22 SBC Billing CDR XML Format Elements (continued)

Element	Attribute	Optional	Description
adjacency	type	N	One of: <ul style="list-style-type: none"> orig (indicates that this adjacency is the originating adjacency of the call) term (indicates that this adjacency is the outgoing adjacency to which the call was routed)
	name	N	Name of the adjacency, as configured by the administrator on the SBC.
	account	N	Name of the account to which the originating or terminating branch of the call belongs, as configured by the administrator on the SBC.
	vpn	Y	VRF name associated with the adjacency (if any).
connect	Time	N	Time the call connected; that is, when the media gate opened.
disconnect	time	N	Time the call disconnected; that is, when the media gate closed.
	reason	N	Reason for the disconnection.
QoS	reservetime	Y	Time QoS was reserved.
	committime	Y	Time that QoS was committed. This must be present if QoS was committed.
	releasetime	N	Time that QoS was released.
gate	No attributes.		
flowinfo	No attributes.		
local	Address	N	IP address that is sending and receiving packets.
	Port	N	Port number that packets are sent from and received on.
remote	Address	N	IP address that is sending and receiving packets.
	Port	N	Port number that packets are sent from and received on.
sd	No attributes.		
RTCPStats	No attributes.		

LDR Format

At a configurable time every 24 hours (usually midnight), the SBE checks the list of currently active calls, and produces a long-duration call record (LDR), with the following structure:

```
<longcall bcid="0123456789" starttime="1110916754000"
          duration="90000000">
  <party type="orig" phone="02083661177"/>
  <party type="term" phone="02083677012"/>
  <adjacency type="orig" name="csi_enfield" account="csi" vpn="0A32F18"/>
  <adjacency type="term" name="softswitch1" account="internal"/>
</longcall>
```



Note

The production of this record is affected by the limitations of cold-failover (see the [“Cold Failover”](#) section).

Partial Call Format

Partial CDRs are generated in the cold-failover case, in which the SBE has no previous knowledge of a call due to catastrophic failure of both the primary and backup system.

When the call is released, as much data as possible is rescued, and the following XML record is generated:

```
<partialcall bcid="01234567890">
  <QoS releasetime = "1110916754000">
    <gate>
      <flowinfo>
        <local address="67.12.43.123" port="46512"/>
        <remote address="67.84.141.2" port="44684"/>
        <sd>
          m=audio 0 RTP/AVP 0
          a=rtpmap:0 PCMU/8000
          aptime:20
        </sd>
        <RTCPstats>
          PS=1245, OS=62345, PR=1362, OR=68095, PD=0, OD=0, PL=0, JI=0, LA=48,
          PC/RPS=1362, PC/ROS=68095, PC/RPR=1245, PC/RPL=0, PC/RJI=0, PC/RLA=33
        </RTCPstats>
      </flowinfo>
      <flowinfo>
        <local address="172.19.8.45" port="48152"/>
        <remote address="172.23.65.41" port="47132"/>
        <sd>
          m=audio 0 RTP/AVP 0
          a=rtpmap:0 PCMU/8000
          aptime:20
        </sd>
        <RTCPstats>
          PS=1362, OS=68095, PR=1245, OR=62345, PD=0, OD=0, PL=0, JI=0, LA=44,
          PC/RPS=1245, PC/ROS=62345, PC/RPR=1362, PC/RPL=0, PC/RJI=0, PC/RLA=31
        </RTCPstats>
      </flowinfo>
    </gate>
  </QoS>
</partialcall>
```

Audit Log Format

Audit logs are generated every hour and summarize the billing activity during the previous hour. An XML fragment follows:

```
<audit time="1110916754000">
  <log>
    <name>billable calls received</name>
    <value>120</value>
  </log>
  <log>
    <name>call records</name>
    <value>100</value>
  </log>
  <log>
    <name>long records</name>
    <value>10</value>
  </log>
  <log>
    <name>partial records</name>
```

```

    <value>5</value>
  </log>
</log>
  <name>lost due to resources</name>
  <value>2</value>
</log>
</log>
  <name>lost due to error</name>
  <value>1</value>
</log>
</audit>

```

Each audit log has the following structure:

- An encompassing <audit> </audit> tag pair, containing the time of the log as an attribute (in milliseconds) since 01-01-1970 midnight UTC.
- A number of loggable parameters, consisting of
 - A name tag, containing the text label for the loggable item
 - A value tag, containing the value of the log.

Table 23 lists the name and value of the log items.

Table 23 Log Item Names and Values

Name	Value range
Billable calls received	Integer, greater than or equal to 0.
Call records	Integer, greater than or equal to 0.
Long records	Integer, greater than or equal to 0.
Partial records	Integer, greater than or equal to 0.
Lost due to resources	Integer, greater than or equal to 0.
Lost due to error	Integer, greater than or equal to 0.

XML DTD

The doctype for the XML follows:

```

<!DOCTYPE recordfile [
  <!ELEMENT recordfile (call | longcall | partialcall | audit)*>
  <!ATTLIST recordfile sbe CDATA #REQUIRED>
  <!ELEMENT call (party, party, adjacency, adjacency, connect?, disconnect?, QoS*)>
  <!ATTLIST call starttime CDATA #REQUIRED
    endtime CDATA #REQUIRED
    duration CDATA #REQUIRED
    bcid CDATA #REQUIRED>
  <!ELEMENT party EMPTY>
  <!ATTLIST party type CDATA #REQUIRED
    phone CDATA #REQUIRED>
  <!ELEMENT adjacency EMPTY>
  <!ATTLIST adjacency type CDATA #REQUIRED
    name CDATA #REQUIRED
    account CDATA #REQUIRED
    vpn CDATA #IMPLIED>
  <!ELEMENT connect EMPTY>
  <!ATTLIST connect time CDATA #REQUIRED>
  <!ELEMENT disconnect EMPTY>

```

```

<!ATTLIST disconnect time CDATA #REQUIRED
                    reason CDATA #REQUIRED>
<!ELEMENT QoS (gate, gate*)>
<!ATTLIST QoS
reservetime CDATA #IMPLIED
                    committime CDATA #IMPLIED
                    releasetime CDATA #IMPLIED>
<!ELEMENT gate (flowinfo, flowinfo)>
<!ELEMENT flowinfo (local, remote, sd, RTPStats)>
<!ELEMENT local EMPTY>
<!ATTLIST local address CDATA #REQUIRED
                    port CDATA #REQUIRED>
<!ELEMENT remote EMPTY>
<!ATTLIST remote address CDATA #REQUIRED
                    port CDATA #REQUIRED>
<!ELEMENT sd (#PCDATA)>
<!ELEMENT RTPblock (#PCDATA)>
<!ELEMENT longcall (party, party)>
<!ATTLIST longcall starttime CDATA #REQUIRED
                    duration CDATA #REQUIRED
                    bcid CDATA #REQUIRED>
<!ELEMENT partialcall (QoS)>
<!ATTLIST partialcall bcid CDATA #REQUIRED>
<!ELEMENT audit (log*)>
<!ELEMENT log (name, value)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT value (#PCDATA)>
]>

```

File Access

The records are appended to a single file, which is periodically flipped and copied to an accessible location, to be retrieved by FTP.

- If the write fails, an alarm is raised.
- If the file size exceeds a preconfigured set of values, an alarm is raised warning the administrator. The severity of the alarm is based on a progression of file size thresholds, ranging from minor to major to critical. The configuration and administration of these alarms are specified in the [“Logging and Alarms”](#) section.

CDR Media Requirements

The requirements to support SBC CDR on nonvolatile media (disk, tape, NVRAM, and so on) are described in the following sections:

- [Space Requirements](#)
- [Performance Requirements](#)

Space Requirements

Each record takes up approximately 1300 bytes. The anticipated number of records is given by:

- 25,000 concurrent calls lasting on average 90 seconds peak time
- Total usage averages 20 percent of peak
- Number of calls in 24 hours; that is, number of records (4,800,000 calls).

Therefore, to store 24 hours of records, the required media space is 6,240,000,000 bytes, or 5.8 Gb. For media space values close to this value, it is imperative that the completed, flipped file is retrieved in a timely fashion, or SBC runs short of space.

Performance Requirements

The performance requirement is that the write rate should be equal to the average time between new call requests in peak hours.

This is given by $90 / 25,000 = 0.0036$ seconds.

Therefore, the write speed must be $1,300 / 0.0036 = 361,000$ bytes/second, or 353 KB/s.

Administration and Configuration

Billing requires the following generic configuration:

- Whether to use standalone or integrated modes of billing (both is not supported).

The remaining configuration is dependent on the operational mode chosen.

Standalone Mode Configuration

If standalone is specified, the following configuration information is required:

- The minor, major, and critical threshold sizes for the CDR file, at which minor, major, and critical alarms are generated.
- The time of day to generate LDRs.

Integrated Mode Configuration

If integrated mode is specified, then the following configuration information is required:

- The assigned element ID. This is an ID assigned by the Internet service provider (ISP). The ID must be unique across the set of SBEs, sending event messages to a particular set of billing servers.
- The minor, major, and critical threshold sizes for the event message cache file.
- The location of the event message cache file on disk.
- The time at which to generate the **Media_Alive** message.
- RADIUS client configuration information.

Integrated mode requires the RADIUS client component of SBC. This has configuration requirements (such as the sets of billing servers). Each of these sets also has a state, which depends on the existence or absence of the event message cache file for that set. The administrator may change this state. The state may be disabled, active, failed, or resending.

Administering SBC Billing

The billing component is administered using the SBC command-line interface. Refer to applicable billing commands in the *Cisco IOS XR Session Border Controller Command Reference*.

The configuration information is modified in real time to effect administration changes. Specifically, the administrator may alter the configuration for a set in the following manner:

- If the state is disabled, then the set is disabled and no event messages are sent to it or cached locally. To enable it, the administrator must set the state to active.
- If the state of a set is active, then there are no failed events, and the administrator cannot do anything except disable it, i.e. change the state to disabled.
- If the state is failed, the set has failed and billing is caching messages for it. The administrator may change the state to either:
 - Resending (which triggers billing to attempt to resend cached messages)
 - Disabled (which deletes the cached event messages and prevents sending of messages).

In this fashion, the administrator may clear the cache and restart a set by setting a server in the failed state to the disabled state, and then to the active state.

It is not possible to change the state directly from the failed state to the active state.

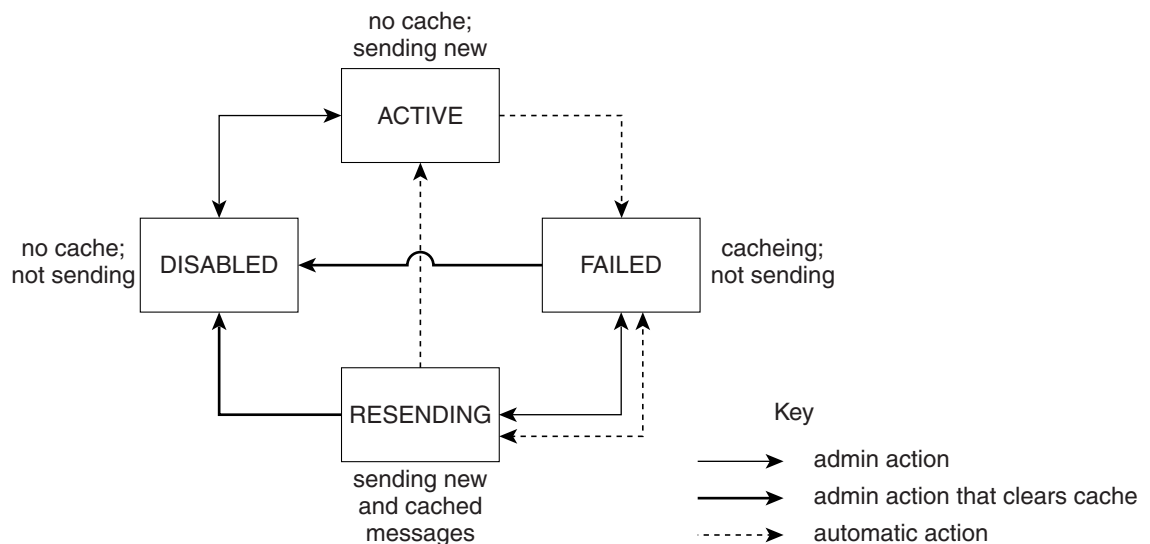
- If the state is resending, then billing is attempting to resend messages from the cache. When this process is complete, the state reverts to the active state. If it fails, it reverts to the failed state. The administrator can change the state to:
 - Failed (which pauses resending)
 - Disabled (which terminates resending and clears the cache)

The administrator may clear the cache by setting the state to the disabled state and then to the active state, to begin sending without resending the cache.

It is not possible to change the state directly from the resending state to the active state.

This process is summarized in [Figure 20](#).

Figure 20 Billing Server Set Administration States



Logging and Alarms

Alarms are tripped differently, based on how billing has been integrated, as described in [Table 24](#).

Table 24 Billing System Logging Conditions

Billing System Type	Logging Conditions
Standalone Billing Alarms	<p>Alarms are tripped under the following conditions:</p> <ul style="list-style-type: none"> • A minor alarm is tripped when the CDR file exceeds a preconfigured minor-threshold value. • A major alarm is tripped when the CDR file exceeds a major-threshold value. • A critical alarm is tripped when the CDR file exceeds a critical-threshold value.
Integrated Billing Alarms	<p>Alarms are tripped under the following conditions:</p> <ul style="list-style-type: none"> • Minor, major, and critical alarms are sent if the cache file size exceeds a preconfigured threshold. • Alarms are tripped when billing servers become unavailable, as follows. <ul style="list-style-type: none"> – A minor alarm is tripped if just one of the configured sets of billing servers is unavailable. – A major alarm is tripped if more than one of the billing server sets is unavailable. – A critical alarm is tripped if none of the billing servers is available. <p>Note In this situation, it may be that the condition for more than one alarm is satisfied (for example, there is just one server set configured, which fails). The most severe alarm dominates.</p>

Fault Tolerance

The SBC billing system is fault tolerant on the following two levels:

- **Warm Failover**—Failover to a live backup (for example, a second card on the same machine).
- **Cold Failover**—Failover to a new machine with no software connection between the defunct machine and the new machine.

Warm Failover

In the event of failover to a backup system, warm failover mechanisms are supported. In the case of warm failover:

- No data is lost on the SBE.
- The value for media statistics for the call on the DBE is reset (this information is lost).

Cold Failover

In the event of failover to a cold, non-dedicated backup, some billing data is lost in the transition from the old, failed system to the new server. The number of billing records completely lost during this transition is less than 10,000 per failover. However, in such a situation, consider the following possibilities:

- The remaining billing records may be corrupted, and only partial billing records recovered. This is especially true with local CDR generation, because no logs are produced in a hard format until the call ends.
- If an event message cache exists on the failed machine, more billing events may be lost, because the disk record may be unrecoverable due to fire, hardware malfunction, or whatever the original cause of the total failure was. This, however, is an unlikely scenario, given that it would require the billing server to be unavailable and unrecovered for a period preceding the cold failover.
- If the media to which the CDRs are written is lost, the entire store of CDRs not backed up (by extracting the records using FTP) is lost.
- It is not possible to detect long-duration calls following a cold failover. Data is only recoverable from the system only when an event occurs in the network, such as the media being terminated).

Security

The *PacketCable 1.5 Event Messages Specification* mandates that the billing messages are sent using the RADIUS protocol and IPSec for security.

**Note**

In Release 3.4.0, only the RADIUS security mechanism, based on its own Request Authenticator, is supported.
