



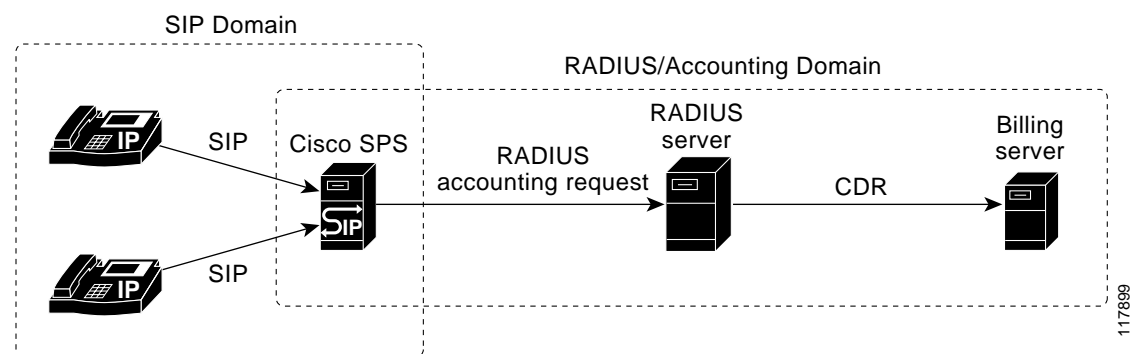
## RADIUS Accounting for Cisco SPS

You can configure Cisco SPS to send Accounting-Request packets to a RADIUS server that correspond to the SIP transactions that the server processes. The Accounting-Request packets are start and stop records that contain a combination of standard RADIUS attributes and Cisco-defined vendor-specific attributes (VSAs).

### RADIUS Accounting for Cisco SPS Overview

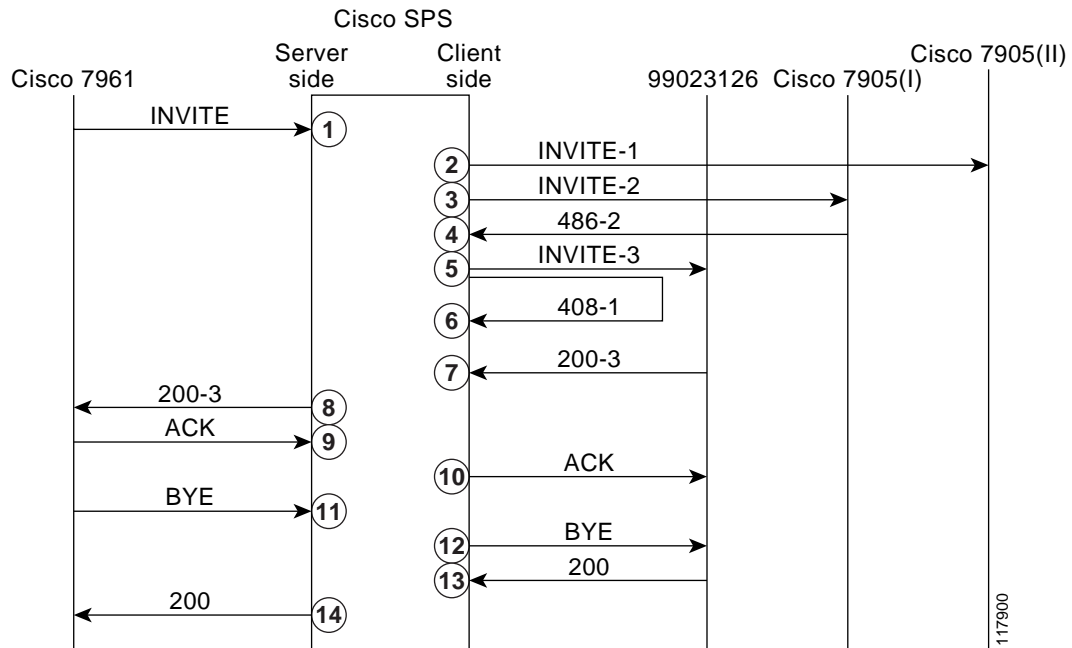
Cisco SPS generates Accounting-Request packets for all branches of each INVITE and BYE transaction. This facilitates start and stop records being sent to a RADIUS server for all call attempts, including all branches of forked call-attempts, whether successful, unsuccessful, or canceled. All Accounting-Request packets for the same call contain the same Call ID, and the RADIUS server, or a billing server working with the RADIUS server, must be able to correlate Accounting-Request packets based on this Call ID. The high-level diagram in [Figure 1](#) shows a SIP call from one SIP phone to another through Cisco SPS and the corresponding RADIUS Accounting-Request packet sent by Cisco SPS to the RADIUS server. In this example, the RADIUS server forms a call detail record (CDR) from the Accounting-Request packet and forwards the CDR to the billing server for correlation.

**Figure 1** Accounting Components



The configuration options in the Cisco SPS allow the customization of accounting triggers. To upstream entities, Cisco SPS appears as a server side entity that handles requests. To downstream entities, Cisco SPS appears as a client side entity that initiates requests. Configuration options enable accounting for call attempts on the server side or client side, as well as for successful and/or unsuccessful call attempts. The call flow in [Figure 2](#) demonstrates the various accounting records that can be generated for a call.

Figure 2 Accounting Call Flow



The following numbered process describes the call-flow protocol outlined in [Figure 2](#):

1. User A wants to call User B. User A sends an INVITE for User B to Cisco SPS.
2. User B has registered at both B1 and B2.
3. Cisco SPS sends the INVITE to both B1 and B2.
4. B2 is busy and returns 486. Cisco SPS sends client side unsuccessful Stop.
5. User B has a Call Forward Busy contact set to B3, so Cisco SPS forwards INVITE to B3.
6. The INVITE to B1 times out. Cisco SPS generates an internal 408 and sends client side unsuccessful Stop.
7. B3 answers the call and returns 200. Cisco SPS sends client side Start.
8. Cisco SPS forwards 200 to User A and sends server side Start.
9. User A sends an ACK
10. Cisco SPS forwards the ACK to User B.
11. User A sends the BYE to Cisco SPS.
12. Cisco SPS forwards the BYE to B3.
13. Cisco SPS receives the 200 for the BYE from B3 and sends client side successful Stop.
14. Cisco SPS forwards the 200 for the BYE to User A and sends server side successful Stop

## RADIUS Server Accounting

For a successful server-side call attempt, a Start record is sent when a 200 final response for the INVITE is returned upstream. A Stop record is sent when a final response is sent for the BYE.

The Start record contains an *h323-start-time*, which is the time the INVITE was received, and an *h323-connect-time*, which is the time the 200 was sent. The *h323-call-origin* is set to *answer*, indicating this is a server-side accounting record. The *sip-status-code* VSA is set to 200, which is the value of the final response for the INVITE. A text representation of the Start, complete with all standard RADIUS attributes and Cisco-defined VSAs, is as follows:

```
NAS-IP-Address = a.4.61.72
NAS-Port-Type = Virtual
User-Name = "1230"
Service-Type = Login-User
Acct-Status-Type = Start
Acct-Session-Id = "04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Called-Station-Id = "<sip:5670@a.4.61.72:5060>"
Calling-Station-Id = "<sip:1230@a.4.61.70:9090>"
Vendor-Specific-9-25 = "h323-setup-time=21:31:14.578 GMT Mon Apr 14 2003"
Vendor-Specific-9-28 = "h323-connect-time=21:31:24.692 GMT Mon Apr 14 2003"
Vendor-Specific-9-26 = "h323-call-origin=answer"
Vendor-Specific-9-27 = "h323-call-type=VoIP"
Vendor-Specific-9-1 = "sip-status-code=200"
Vendor-Specific-9-1 = "session-protocol=sip"
Vendor-Specific-9-1 = "call-id=04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Vendor-Specific-9-1 = "method=INVITE"
Vendor-Specific-9-1 = "prev-hop-via=SIP/2.0/UDP a.4.61.70:9090"
Vendor-Specific-9-1 = "prev-hop-ip=a.4.61.70:9090"
Vendor-Specific-9-1 = "incoming-req-uri=sip:5670@a.4.61.72:5060"
Vendor-Specific-9-1 = "outgoing-req-uri=sip:5670@a.4.106.19:5060"
Vendor-Specific-9-1 = "next-hop-ip=a.4.106.19:5060"
```

The Stop record contains an *h323-disconnect-time*, which is the time the BYE was received. The *h323-call-origin* is set to *answer*, indicating this is a server-side accounting record. The *h323-disconnect-cause* is not used; instead, the *sip-status-code* VSA is added and set to the value of the final response for the BYE. For this reason, the Stop is not sent until the final response for the BYE has been sent. A text representation of the Stop, complete with all standard RADIUS attributes and Cisco-defined VSAs, is as follows:

```
NAS-IP-Address = a.4.61.72
NAS-Port-Type = Virtual
User-Name = "1230"
Service-Type = Login-User
Acct-Status-Type = Stop
Acct-Session-Id = "04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Called-Station-Id = "<sip:5670@a.4.61.72:5060>;tag=1F37F280-21AD"
Calling-Station-Id = "<sip:1230@a.4.61.70:9090>"
Vendor-Specific-9-29 = "h323-disconnect-time=21:31:44.770 GMT Mon Apr 14 2003"
Vendor-Specific-9-26 = "h323-call-origin=answer"
Vendor-Specific-9-27 = "h323-call-type=VoIP"
Vendor-Specific-9-1 = "sip-status-code=200"
Vendor-Specific-9-1 = "session-protocol=sip"
Vendor-Specific-9-1 = "call-id=04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Vendor-Specific-9-1 = "method=BYE"
Vendor-Specific-9-1 = "prev-hop-via=SIP/2.0/UDP a.4.61.70:9090"
Vendor-Specific-9-1 = "prev-hop-ip=a.4.61.70:9090"
Vendor-Specific-9-1 = "incoming-req-uri=sip:5670@a.4.61.72:5060"
Vendor-Specific-9-1 = "outgoing-req-uri=sip:5670@a.4.106.19:5060"
Vendor-Specific-9-1 = "next-hop-ip=a.4.106.19:5060"
```

For an unsuccessful server-side call attempt, there is no Start record. A Stop record is sent when the best non-200 response for the INVITE is returned upstream, including the CANCEL scenario in which Cisco SPS waits for the 487 from the downstream and returns it upstream. The Stop record contains an *h323-start-time*, which is the time the INVITE message was received, and an *h323-disconnect-time*,

which is the time the final response was sent. The h323-call-origin is set to answer, indicating this is a server-side accounting record. The sip-status-code VSA is set to the value of the final response for the INVITE. A text representation of the Stop, complete with all standard RADIUS attributes and Cisco-defined VSAs, is as follows:

```
NAS-IP-Address = a.4.61.72
NAS-Port-Type = Virtual
User-Name = "1230"
Service-Type = Login-User
Acct-Status-Type = Stop
Acct-Session-Id = "04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Called-Station-Id = "<sip:5670@a.4.61.72:5060>;tag=1F37F280-21AD"
Calling-Station-Id = "<sip:1230@a.4.61.70:9090>"
Vendor-Specific-9-25 = "h323-setup-time=21:31:14.578 GMT Mon Apr 14 2003"
Vendor-Specific-9-29 = "h323-disconnect-time=21:31:44.770 GMT Mon Apr 14 2003"
Vendor-Specific-9-26 = "h323-call-origin=answer"
Vendor-Specific-9-27 = "h323-call-type=VoIP"
Vendor-Specific-9-1 = "sip-status-code=487"
Vendor-Specific-9-1 = "session-protocol=sip"
Vendor-Specific-9-1 = "call-id=04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Vendor-Specific-9-1 = "method=INVITE"
Vendor-Specific-9-1 = "prev-hop-via=SIP/2.0/UDP a.4.61.70:9090"
Vendor-Specific-9-1 = "prev-hop-ip=a.4.61.70:9090"
Vendor-Specific-9-1 = "incoming-req-uri=sip:5670@a.4.61.72:5060"
Vendor-Specific-9-1 = "outgoing-req-uri=sip:5670@a.4.106.19:5060"
Vendor-Specific-9-1 = "next-hop-ip=a.4.106.19:5060"
```

## RADIUS Client Accounting

For the client-side accounting, Accounting-Request packets (Starts or Stops) are sent on behalf of each branch when a final response is received on that branch. If the final response is a 200 for an INVITE, a Start record is sent for that client-side branch. The corresponding Stop record is sent when the final response is received or generated for the BYE. If the final response for the INVITE is a non-200 response, a Stop is generated for the given client-side branch when the final response is received or generated.

For a successful client-side call, the Start record contains an h323-start-time, which is the time the INVITE was sent, and an h323-connect-time, which is the time the 200 was received. The h323-call-origin is set to originate indicating this is a client-side accounting record. The sip-status-code VSA is set to 200, which is the value of the final response for the INVITE. A text representation of a sample Start, complete with all standard RADIUS attributes and Cisco defined VSAs, is as follows:

```
NAS-IP-Address = a.4.61.72
NAS-Port-Type = Virtual
User-Name = "1230"
Service-Type = Login-User
Acct-Status-Type = Start
Acct-Session-Id = "04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Called-Station-Id = "<sip:5670@a.4.61.72:5060>"
Calling-Station-Id = "<sip:1230@a.4.61.70:9090>"
Vendor-Specific-9-25 = "h323-setup-time=21:31:14.578 GMT Mon Apr 14 2003"
Vendor-Specific-9-28 = "h323-connect-time=21:31:24.692 GMT Mon Apr 14 2003"
Vendor-Specific-9-26 = "h323-call-origin=originate"
Vendor-Specific-9-27 = "h323-call-type=VoIP"
Vendor-Specific-9-1 = "sip-status-code=200"
Vendor-Specific-9-1 = "session-protocol=sip"
Vendor-Specific-9-1 = "call-id=04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Vendor-Specific-9-1 = "method=INVITE"
```

```

Vendor-Specific-9-1 = "prev-hop-via=SIP/2.0/UDP a.4.61.70:9090"
Vendor-Specific-9-1 = "prev-hop-ip=a.4.61.70:9090"
Vendor-Specific-9-1 = "incoming-req-uri=sip:5670@a.4.61.72:5060"
Vendor-Specific-9-1 = "outgoing-req-uri=sip:5670@a.4.106.19:5060"
Vendor-Specific-9-1 = "next-hop-ip=a.4.106.19:5060"

```

When a final response for a BYE is received or generated, a Stop record is sent on behalf of the client. The Stop record contains an h323-disconnect-time, which is the time the final response was received or generated. The h323-call-origin is set to originate, indicating this is a client-side accounting record. The h323-disconnect-cause is not used; instead, the sip-status-code VSA is added and set to the value of the final response for the BYE. For this reason, the Stop is not sent until the final response for the BYE has been received. A text representation of a sample Stop, complete with all standard RADIUS attributes and Cisco defined VSAs, is as follows:

```

NAS-IP-Address = a.4.61.72
NAS-Port-Type = Virtual
User-Name = "1230"
Service-Type = Login-User
Acct-Status-Type = Stop
Acct-Session-Id = "04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Called-Station-Id = "<sip:5670@a.4.61.72:5060>;tag=1F37F280-21AD"
Calling-Station-Id = "<sip:1230@a.4.61.70:9090>"
Vendor-Specific-9-29 = "h323-disconnect-time=21:31:44.770 GMT Mon Apr 14 2003"
Vendor-Specific-9-26 = "h323-call-origin=originate"
Vendor-Specific-9-27 = "h323-call-type=VoIP"
Vendor-Specific-9-1 = "sip-status-code=200"
Vendor-Specific-9-1 = "session-protocol=sip"
Vendor-Specific-9-1 = "call-id=04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Vendor-Specific-9-1 = "method=BYE"
Vendor-Specific-9-1 = "prev-hop-via=SIP/2.0/UDP a.4.61.70:9090"
Vendor-Specific-9-1 = "prev-hop-ip=a.4.61.70:9090"
Vendor-Specific-9-1 = "incoming-req-uri=sip:5670@a.4.61.72:5060"
Vendor-Specific-9-1 = "outgoing-req-uri=sip:5670@a.4.106.19:5060"
Vendor-Specific-9-1 = "next-hop-ip=a.4.106.19:5060"

```

An unsuccessful Stop record is generated for any client-side branch for which a non-200 response for the INVITE is received or generated. When a CANCEL is sent down a branch, Cisco SPS waits for the 487, for the corresponding INVITE from downstream to send a Stop record, indicating this branch was cancelled. The Stop record contains an h323-start-time, which is the time the INVITE was sent, and an h323-disconnect-time, which is the time the final response was received or generated. The h323-call-origin is set to originate indicating this is a client accounting record. The sip-status-code VSA is set to the value of the final response for the INVITE. A text representation of a sample Stop, complete with all standard RADIUS attributes and Cisco defined VSAs, is as follows:

```

NAS-IP-Address = a.4.61.72
NAS-Port-Type = Virtual
User-Name = "1230"
Service-Type = Login-User
Acct-Status-Type = Stop
Acct-Session-Id = "04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Called-Station-Id = "<sip:5670@a.4.61.72:5060>;tag=1F37F280-21AD"
Calling-Station-Id = "<sip:1230@a.4.61.70:9090>"
Vendor-Specific-9-25 = "h323-setup-time=21:31:14.578 GMT Mon Apr 14 2003"
Vendor-Specific-9-29 = "h323-disconnect-time=21:31:44.770 GMT Mon Apr 14 2003"
Vendor-Specific-9-26 = "h323-call-origin=originate"
Vendor-Specific-9-27 = "h323-call-type=VoIP"
Vendor-Specific-9-1 = "sip-status-code=487"
Vendor-Specific-9-1 = "session-protocol=sip"
Vendor-Specific-9-1 = "call-id=04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Vendor-Specific-9-1 = "method=INVITE"

```

```

Vendor-Specific-9-1 = "prev-hop-via=SIP/2.0/UDP a.4.61.70:9090"
Vendor-Specific-9-1 = "prev-hop-ip=a.4.61.70:9090"
Vendor-Specific-9-1 = "incoming-req-uri=sip:5670@a.4.61.72:5060"
Vendor-Specific-9-1 = "outgoing-req-uri=sip:5670@a.4.106.19:5060"
Vendor-Specific-9-1 = "next-hop-ip=a.4.106.19:5060"

```

## sip-hdr VSA

Cisco SPS provides configuration options with which you can specify any SIP headers to be sent in VSA 1 (AVPair) within RADIUS Accounting-Request packets. The headers included in the various requests are extracted from SIP messages as follows:

- Server-side Start: 200 for the INVITE returned upstream
- Server-side Stop (unsuccessful call): 3xx-6xx for the INVITE returned upstream
- Server-side Stop (successful call): Final response for the BYE received or generated
- Client-side Start: 200 for INVITE received from downstream branch
- Client-side Stop (unsuccessful branch): 3xx-6xx for the INVITE received (or generated)
- Client-side Stop (successful branch): Final response for the BYE that was sent

A text representation of a sample Start in which Cisco SPS has been configured to include the CSeq and Contact headers, complete with all standard RADIUS attributes and Cisco defined VSAs, is as follows (note that the sip-hdr attributes appear in bold for illustrative purposes):

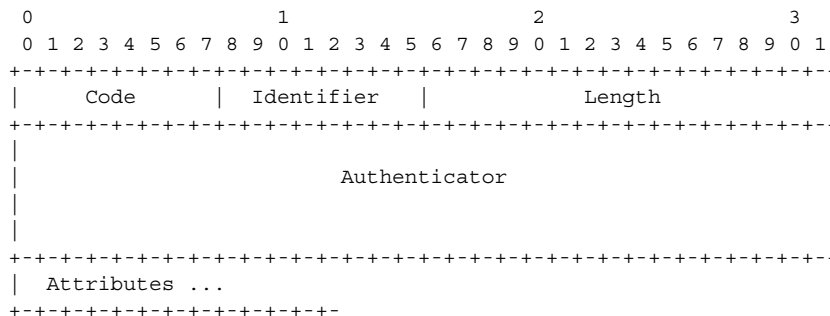
```

NAS-IP-Address = a.4.61.72
NAS-Port-Type = Virtual
User-Name = "1230"
Service-Type = Login-User
Acct-Status-Type = Start
Acct-Session-Id = "04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Called-Station-Id = "<sip:5670@a.4.61.72:5060>"
Calling-Station-Id = "<sip:1230@a.4.61.70:9090>"
Vendor-Specific-9-25 = "h323-setup-time=21:31:14.578 GMT Mon Apr 14 2003"
Vendor-Specific-9-28 = "h323-connect-time=21:31:24.692 GMT Mon Apr 14 2003"
Vendor-Specific-9-26 = "h323-call-origin=answer"
Vendor-Specific-9-27 = "h323-call-type=VoIP"
Vendor-Specific-9-1 = "sip-status-code=200"
Vendor-Specific-9-1 = "session-protocol=sip"
Vendor-Specific-9-1 = "call-id=04fb5d3908f3bfbe24fabf24f9bfbe@a.4.61.70"
Vendor-Specific-9-1 = "method=INVITE"
Vendor-Specific-9-1 = "prev-hop-via=SIP/2.0/UDP a.4.61.70:9090"
Vendor-Specific-9-1 = "prev-hop-ip=a.4.61.70:9090"
Vendor-Specific-9-1 = "incoming-req-uri=sip:5670@a.4.61.72:5060"
Vendor-Specific-9-1 = "outgoing-req-uri=sip:5670@a.4.106.19:5060"
Vendor-Specific-9-1 = "next-hop-ip=a.4.106.19:5060"
Vendor-Specific-9-1 = "sip-hdr=Contact: <sip:1230@a.4.61.70:5060>"
Vendor-Specific-9-1 = "sip-hdr=CSeq: 101 INVITE"

```

# RADIUS Data Format

A summary of the RADIUS data format, as defined in RFC 2866 follows. The RADIUS accounting record fields are transmitted from left to right.



The following sections describe the specific fields that make up the accounting record.

## Code

The *Code* field is one octet and identifies the type of RADIUS packet. RADIUS accounting codes (decimal) are assigned as follows:

- 4—Accounting-Request code (used in Start and Stop requests sent by Cisco SPS)
- 5—Accounting-Response code (used in Start and Stop responses returned by the RADIUS server)

Upon receipt of an Accounting-Request packet, the RADIUS server *must* transmit an Accounting-Response packet reply if the server successfully records the accounting packet—and *must not* transmit any reply if the server fails to record the accounting packet.

## Identifier

The *Identifier* field is one octet and aids in matching requests and replies. The RADIUS server can detect a duplicate request if the server has the same client source IP address and source UDP port and Identifier field within a short span of time. Cisco SPS changes the Identifier field whenever the content of the Attributes field changes and whenever a valid reply has been received for a previous request. For retransmissions where the contents are identical, the Identifier field remains unchanged. Cisco SPS uses monotonically increasing identifiers that wrap around to 1 once they reach 254.

## Length

The *Length* field is two octets and indicates the length of the packet including the Code, Identifier, Length, Authenticator, and Attribute fields. Octets outside the range of the Length field *must* be treated as padding and ignored on reception. If a packet is shorter than the Length field indicates, the packet *must* be silently discarded. The minimum length of the RADIUS Accounting Record packet is 20 and maximum length is 4095. If Cisco SPS is configured to include more headers than fit within the maximum length, the Accounting-Request packet is truncated.

## Authenticator

The *Authenticator* field is sixteen (16) octets. The most significant octet is transmitted first. The Authenticator field value is used to authenticate the messages between the client and the RADIUS accounting server.

## Request Authenticator

In Accounting-Request packets, the Authenticator value is a 16-octet Message Digest 5 (MD5) checksum, which is called the *Request Authenticator*. Cisco SPS and the RADIUS server share a configurable secret. The Request Authenticator field in Accounting-Request packets contains a one-way MD5 hash calculated over a stream of octets, consisting of the Code + Identifier + Length + 16 zero octets + request attributes + shared secret (where + indicates concatenation). The 16-octet MD-hash value is stored in the Authenticator field of the Accounting-Request packet. The shared secret is configured on Cisco SPS as described in [“Accounting Module Directives” section on page 15](#).

## Response Authenticator

The Authenticator field in an Accounting-Response packet is called the *Response Authenticator*, and contains a one-way MD5 hash that is calculated over a stream of octets, consisting of the Accounting-Response Code, Identifier, Length, the Request Authenticator field from the Accounting-Request packet being replied to, and the response attributes if any, followed by the shared secret. The resulting 16-octet MD5-hash value is stored in the Authenticator field of the Accounting-Response packet. The shared secret that is configured on the RADIUS server *must* match the shared secret that is configured on Cisco SPS.

## Attributes

The *Attributes* field is variable in length and contains a list of Attributes fields. Attributes may have multiple instances. The end of the list of attributes is indicated by the Length field of the RADIUS packet. A summary of the attribute format follows. The fields are transmitted from left to right.

```

0                               1                               2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Type   |   Length   |   Value ...
+-----+-----+-----+-----+-----+-----+

```

## Standard Attributes

The standard RADIUS attributes in RFC 2865 and RFC 2866 that are included in Accounting-Request packets from Cisco SPS are listed in [Table 1](#). The Attribute column is for illustrative purposes only because it is not a field in the actual attributes that are included in the request. The *Length* field is in octets and is 1 (for the *Type* field) + 1 (for the *Length* field) + length of the *Value* field.

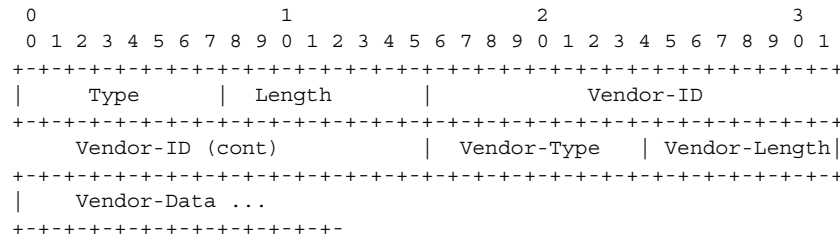


**Table 1** Standard Attributes for Accounting-Request Packets

Attribute	Type	Length (octets)	Value
User Name	1	3-254	<p>User part of the URL in the “From” SIP header or user in “Proxy-Authorization” SIP header, if present. This value is controlled by a configuration directive (OrigUserNameSource).</p> <p>If the RadiusUserNameAttrAddDomain directive is On, the domain in the “From” header will be appended to the User Name and the User Name attribute will be in the name@domain format.</p> <p>The User Name can be expanded to a fully expanded E164 number. If the user name is taken from the “From” header, the expansion is dependent on the user type and the NumericUserNameInterpretation directive. If the user name is taken from the “Proxy-Authorization” header, the expansion is dependent on the NumExpandAuthUserName directive. The expansion is based on the number expansion rules, which are defined in the number expansion module.</p> <p>Example: “5000”</p> <p>Example: “+1xxxxyy1212”</p> <p>Example: “bob@foo.com”</p>
NAS IP	4	6	<p>IP address of Cisco SPS.</p> <p>Example: a.22.76.1</p>
Service-Type	6	6	<p>1, which stands for Login.</p> <p>Example: 1</p>
Vendor-Specific	26	7-254	See <a href="#">Table 2</a> for the complete set.
Called-Station-Id	30	3-254	<p>“To” SIP header value of the request.</p> <p>Example: "&lt;sip:5004@cisco.com&gt;; tag=c3943000dd3-ba2b824"</p>
Calling-Station-Id	31	3-254	<p>“From” SIP header value of the request.</p> <p>Example: "&lt;sip:5003@cisco.com&gt;;tag=c1234567dd3-ab2b248"</p>
Acct-Status-Type	40	6	<p>1 for Start, 2 for Stop.</p> <p>Example: 1</p>
Acct-Session-Id	44	3-254	<p>“Call-Id” SIP header value of the request.</p> <p>Example: "003094c3-ad5f5e93-2351ccb4-436f323d@a.22.76.38"</p>
NAS Port Type	61	6	<p>5 stands for Virtual. Virtual refers to a connection by the client to Cisco SPS via a transport protocol. In this case, SIP is used, instead of a physical port.</p> <p>Example: 5</p>

## Vendor-Specific Attributes

A summary of the vendor-specific attribute (VSA) format is as follows:



The values of these fields for Accounting-Request packets from Cisco SPS are the following:

- Type = 26 (indicating Vendor-Specific)
- Length = 7-254 octets
- Vendor-ID = 9 (Cisco's Vendor-ID for all VSAs used by Cisco SPS)
- Vendor-Type, Vendor-Length, and Vendor-Data take various values as summarized in [Table 2](#).

The Attribute column is for illustrative purposes only because it is not a field in the actual attributes that are included in the request. The Vendor-Length field is in octets and is 1 (for the Vendor-Type field) + 1 (for the Vendor-Length field) + length of the Vendor-Data field. For additional details on all Cisco VSAs, see [http://www.cisco.com/univercd/cc/td/doc/product/access/acs\\_serv/vapp\\_dev/vsaig3.htm#27593](http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/vsaig3.htm#27593).

**Table 2** Cisco VSAs for Accounting Request packets

Attribute	Vendor Type	Vendor Length (octets)	Vendor Data
h323-conf-id	24	Less than or equal to 247	“cisco-GUID” SIP header value of the request. If this header is not present in the SIP request, this VSA is omitted from the accounting request. Example: “h323-conf-id=2176014509-3094090198-2164699425-3804311776”
h323-call-setup-time	25	Less than or equal to 247	Time the SIP INVITE was processed; GMT format. This VSA is omitted if the accounting request is not for a SIP INVITE. Example: “h323-setup-time=21:31:14.578 GMT Mon Apr 14 2003”
h323-call-origin	26	Less than or equal to 247	<i>answer</i> for server side, or <i>originate</i> of client side. Example: “h323-call-origin=answer”
h323-call-type	27	Less than or equal to 247	VoIP. Example: “h323-call-type=VoIP”
h323-connect-time	28	Less than or equal to 247	Time the response for SIP INVITE was processed; GMT format. This VSA is included in Start requests only. Example: “h323-connect-time=21:31:24.692 GMT Mon Apr 14 2003”

Table 2 Cisco VSAs for Accounting Request packets

Attribute	Vendor Type	Vendor Length (octets)	Vendor Data
h323-disconnect-time	29	Less than or equal to 247	Time the response was processed for call-ending request; this may be a non-200 response for an INVITE or a response for a BYE. This VSA is included in Stop requests only. Example: "h323-disconnect-time=21:31:44.770 GMT Mon Apr 14 2003"
session-protocol	1	Less than or equal to 247	sip. Example: "session-protocol=sip"
Call-id	1	Less than or equal to 247	"Call-Id" SIP header value. Example: "call-id=003094c3-ad5f5e93-2351ccb4-436f323d@a.22.76.38"
Method	1	Less than or equal to 247	Method name given in the request line which may be INVITE or BYE. Start requests always have INVITE, but Stop requests have either INVITE (for an unsuccessful call attempt) or BYE. Example: "method=INVITE"
incoming-req-uri	1	Less than or equal to 247	Request URI of the incoming SIP request. Example: "incoming-req-uri=sip:5004@b.19.174.99"
prev-hop-ip	1	Less than or equal to 247	Previous hop IP address as seen by the proxy. Example: "prev-hop-ip=a.22.76.38:5060"
prev-hop-via	1	Less than or equal to 247	The topmost "Via" header when the SIP request is received by Cisco SPS. Example: "prev-hop-via=SIP/2.0/UDP a.22.76.38:5060;received=a.22.76.38"
outgoing-req-uri	1	Less than or equal to 247	Request URI of the outgoing SIP request sent by Cisco SPS. Example: "outgoing-req-uri=sip:5004@a.22.76.14:5060"
next-hop-ip	1	Less than or equal to 247	IP to which the SIP request is forwarded by Cisco SPS. Example: "next-hop-ip=a.22.76.104:5060"
next-hop-dn	1	Less than or equal to 247	Domain name or fully qualified domain name where the SIP request is forwarded by Cisco SPS. This VSA is omitted if the next-hop domain name or fully qualified domain name is not known. Example: "next-hop-dn=cisco.com"
sip-hdr	1	Less than or equal to 247	An arbitrary SIP header found in the SIP request received by Cisco SPS (complete header line). Inclusion of any given header is controlled by a configuration directive AcctIncludeSIPHeader. Example: "sip-hdr=CSeq: 101 INVITE" Example: "sip-hdr=Via: SIP/2.0/UDP b.19.174.99:5060;branch=363e586d-e1529570-dad25df7-49a3df73-1,SIP/2.0/UDP a.22.76.38:5060;received=a.22.76.38"

## Correlating Accounting Records from Cisco SPS

If both server-side and client-side accounting are enabled, all successful calls result in a Start and a Stop being sent for both the server side and the client side. Enabling server-side and client-side accounting also results in 0 or more Stops being sent for any client-side branches which were not successful. All unsuccessful calls result in a Stop being sent for the server side, and a Stop being sent for any client-side branches. It is possible that there are no client-side branches (such as when a translation phase yields no destinations for the call). All Starts and Stops corresponding to the same call can be correlated based on the fact they all contain the same Call-Id. The Call-Id appears in the Start and Stop as both the Acct-Session-Id (see [Table 1](#)) and the *call-id* VSA (see [Table 2](#)).

Accounting-Request packets for individual client side branches may be distinguished by examining the “To” tag, which is contained in the Called-Station-Id attribute (see [Table 1](#)) associated with each record.

```
Called-Station-Id = "<sip:5670@a.4.61.72:5060>;tag=1F37F280-21AD"
```

In the example above, the tag value is 1F37F280-21AD. Any record for a specific client-side branch has the same tag in addition to having the same Call-Id. For successful calls, a single client-side Stop has the same tag in the Called-Station-Id attribute as the previously sent client-side Start. The server side Start and Stop share this same tag. Because the BYE request may be sent from either side of the call, the “To” tag of the Start may instead match the “From” tag, which is contained in the Calling-Station-Id attribute (see [Table 1](#)) of the Stop.

```
Called-Station-Id = "<sip:5670@a.4.61.72:5060>;tag=1F37F280-21AD"
```

Therefore, if the “To” tag of the Stop does not match the “To” tag of any Starts, the “From” tag of the Stop must be matched against the “To” tag of the Starts as well.

## Re-INVITES

A re-INVITE for the same call results in another set of Accounting-Request packets with the same Call-Id. In order to distinguish the original INVITE from each re-INVITE, the CSeq header may be included in the accounting records as a *sip-hdr* VSA (see [Table 2](#)).

```
Vendor-Specific-9-1 = "sip-hdr=CSeq: 101 INVITE"
```

The CSeq value is different and increases for each re-INVITE. For example, the Starts for the original INVITE could all have CSeq equal to 101. The Starts for the first re-INVITE would have CSeq equal to 102. The Starts for the next re-INVITE would have CSeq equal to 103. This continues until the call ends with a BYE message when the Stops are sent with CSeq equal to the last re-INVITE plus one, or 104. Optionally, the extra Start records for any re-INVITES can be dismissed as duplicates.

## Server-Side and Client-Side Accounting Records

The value of the h323-call-origin VSA (see [Table 2](#)) distinguishes server-side accounting requests from client-side accounting requests.

```
Vendor-Specific-9-26 = "h323-call-origin=answer"
```

The h323-call origin VSA has the value of answer for the server side and originate for the client side.

## Duplicate Accounting Records Caused by Proxy Sending to Itself

Logic exists in Cisco SPS blocking accounting requests when Cisco SPS receives and sends accounting requests from or to itself in order to avoid generating duplicate Accounting-Request packets. This logic includes Cisco SPS recognizing SIP messages from and to the IP address associated with any farm member as being from or to itself.

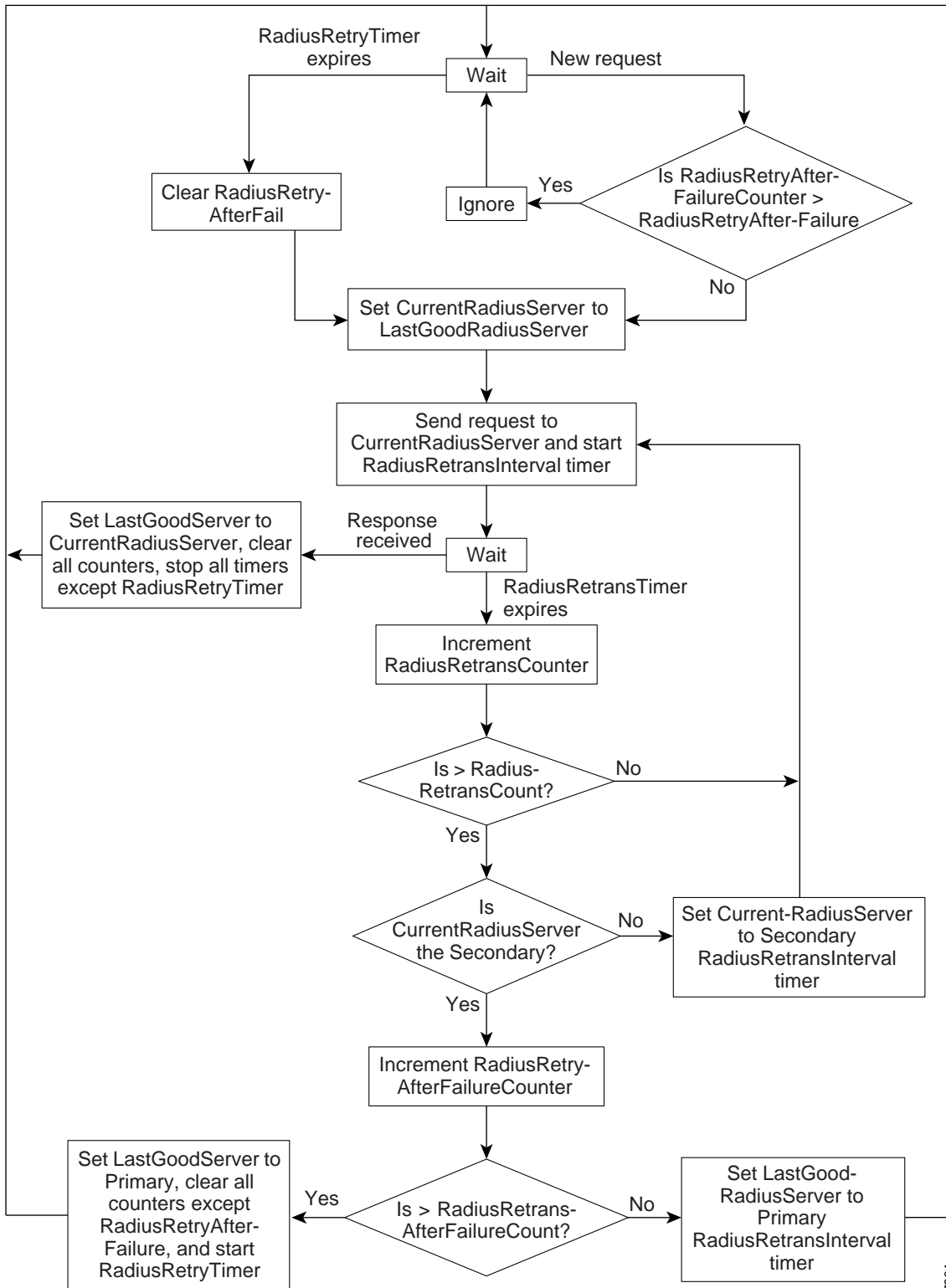
## Configuring RADIUS Accounting

You can configure Cisco SPS to send Accounting-Request packets to up to two RADIUS servers. For initial testing, it may be more convenient to set up a single RADIUS server only; but in a production environment, using a pair of RADIUS servers for redundancy is strongly recommended.

Cisco SPS has multiple SIP message-handling processes (sipds). Each process handles individual SIP messages. In the event the given SIP message is one that requires accounting, it is the responsibility of the sipd handling the message to send the corresponding Accounting-Request packet. The Accounting-Request packet is generated and sent after the SIP message processing has been completed. This means that SIP request processing is not delayed or dependent on accounting.

The algorithm by which the sipd selects a RADIUS server to which to send the Accounting-Request packet is shown in [Figure 3](#).

Figure 3 RADIUS Request State Machine



117901

The values of the various retransmission timers and counters, as well as the configuration of the primary and the secondary RADIUS servers, are described in two sections:

- [Accounting Module Directives, page 15](#)
- [Accounting-Related SIP Server Core Directives, page 16.](#)

## Accounting Module Directives

The following directives are specific to the Cisco SPS accounting module. Reasonable default values are assigned when possible; however, it is advisable to carefully set and review the values of all these directives.



### Note

Complete details regarding viewing and editing Cisco SPS directives are beyond the scope of RADIUS Interface for Cisco SPS. Please refer to the Cisco SIP Proxy Server Administrator Guide for comprehensive configuration information. See the following URL:

<http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/admingd/index.htm>

The following notes summarize how directives are used and how to choose appropriate values for each directive within the context of a RADIUS-based environment:

- **Accounting**—Turns on or off Cisco SPS accounting requests. Valid values are On and Off. When Off, no accounting requests are sent and all the subsequent directives are ignored. The default is Off. For accounting to work properly, this directive must be set to On.
- **AccountingServerSide**—Turns on or off server-side accounting requests for successful calls. Valid values are On and Off. The default is On.
- **AccountingClientSide**—Turns on or off client-side accounting requests for successful calls. Valid values are On and Off. The default is Off.
- **AccountingUnsuccessful**—Turns on or off accounting requests for unsuccessful calls. Valid values are On and Off.

When the flag is On, this directive is interpreted in conjunction with the AccountingServerSide and AccountingClientSide directives.

If the AccountingServerSide and this flag are both On, accounting requests are sent for server-side unsuccessful calls.

If the AccountingClientSide and this flag are both On, accounting records are sent for client-side unsuccessful calls.

When the flag is Off, no accounting requests are sent for unsuccessful calls, regardless of the setting for the AccountingServerSide or AccountingClientSide flags.

The default is Off.

- **AccountingRecordFormat**—Record format used for accounting. Currently, RADIUS is the only valid option.
- **AccountingTimeFormat**—Timestamps are in GMT format.
- **PrimaryRadiusAcctIp**—IP address or host name of the primary RADIUS server to be used for accounting. The default address is 127.0.0.1. For accounting to work properly, the address must be set to an IP address of a functioning RADIUS server.

- **PrimaryRadiusAcctPort**—Destination port number of the primary RADIUS server to be used for accounting. The well-known RADIUS server port for accounting is 1813. The default is 0. For accounting to work properly, set this directive to the port on which the primary RADIUS server is listening for Accounting-Request packets.
- **PrimaryRadiusAcctSecret**—Secret text string shared between Cisco SPS and the primary RADIUS accounting server. This is used to compute the Authenticator field of the Accounting-Request packets as described in the “Request Authenticator” section on page 8. For accounting to work properly, set this directive to the same value that is configured on the RADIUS server.
- **SecondaryRadiusAcctIp**—IP address or host name of the secondary RADIUS server to be used for accounting. The default is 127.0.0.1. For accounting to work properly, set this directive to IP address of a functioning RADIUS server.
- **SecondaryRadiusAcctPort**—Destination port number of the secondary RADIUS server to be used for accounting. The well known RADIUS server port for accounting is 1813. The default port value is 0. For accounting to work properly, set this directive to the port on which the secondary RADIUS server is listening for Accounting-Request packets.
- **SecondaryRadiusAcctSecret**—Secret text string shared between Cisco SPS and the secondary RADIUS accounting server. This text string is used to compute the Authenticator field of the Accounting-Request packets as described in “Request Authenticator” section on page 8. For accounting to work properly, set this directive to the same value that is configured on the RADIUS server.
- **AcctIncludeSIPHeader**—SIP header to be sent as a sip-hdr VSA (see Table 2) within RADIUS Accounting-Request packets. You can configure a maximum of 50 headers.

## Accounting-Related SIP Server Core Directives

The following directives reside in the SIP server core module, but these directives affect the accounting operation of Cisco SPS. These directives are shared across the accounting, authentication, and preauthentication modules. Reasonable default values are assigned when possible; however, it is advisable to carefully set and review the values of all these directives.



### Note

Complete details regarding viewing and editing Cisco SPS directives are beyond the scope of *RADIUS Interface for Cisco SPS*. Please refer to the *Cisco SIP Proxy Server Administrator Guide* for comprehensive configuration information. See the following URL:

<http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/admingd/index.htm>

The following notes summarize how directives are used and how to choose appropriate values for each within the context of a RADIUS-based environment:

- **StatefulServer**—Determines whether Cisco SPS is a transaction-stateful or transaction-stateless server. A transaction includes the following: received request, request or requests (if forked) forwarded downstream, responses received from downstream hosts, and best response returned upstream. Valid values are as follows:
  - On (stateful)—Cisco SPS remembers incoming and outgoing requests, provides reliable retransmission of proxied requests, and returns the best final responses.
  - Off (stateless)—Cisco SPS forgets all information once a request or response has been processed. Cisco SPS forwards requests and responses.



The default is On. If you change the value of this directive, you must restart the server. For accounting to work properly, set this directive to On. If Off, Accounting will still be performed, but not all the attributes are written in the individual accounting requests. The `StatefulServer` directive is assumed to be On for all subsequent configuration requirements described in *RADIUS Interface for Cisco SPS*.

- `ServerType`—Determines whether Cisco SPS functions as a proxy server or as a redirect server. A proxy server processes and routes SIP requests. A redirect server provides contact information by means of SIP redirect (3xx) responses. Valid values are Proxy and Redirect. The default is Proxy. For client-side accounting to work properly, set this directive to Proxy. If `ServerType` is set to Redirect, only server-side accounting is performed.
- `OrigUserNameSource`—Origin of the User Name attribute (see to [Table 1 on page 9](#)) in Accounting-Request packets. Valid values are From and Auth. If the value is From, the user part of the URL in the From SIP header is used to populate the User Name attribute. If the value is Auth, the user provided for authentication in the Authorization or Proxy-Authorization header is used instead. If no Proxy-Authorization header is present, the user is taken from the From header regardless of the setting of this directive. The default is Auth. A relevant configuration for this directive is one where all the IP phones authenticating themselves to Cisco SPS share a common user and password for authentication purposes. For example, Bob and Mary both work for Cisco and populate the From header with Bob and Mary respectively. However, they authenticate themselves to Cisco SPS by using a common username of cisco and a common password. The `OrigUserNameSource` provides to ability to choose whether to have Bob or Mary versus cisco as the User-Name in Accounting-Request packets.
- `AddRecordRoute`—Adds the Record-Route header to an initial SIP INVITE message. The Record-Route header field contains a globally reachable Request-URI that identifies the proxy server that added it. When a proxy server adds a Record-Route header in a SIP INVITE request, the proxy server is kept in the path of subsequent requests for the same call (such as re-INVITES, ACK, BYE). Valid values are On (add) and Off (do not add). The default is Off. `ServerType` must be set to Proxy for this directive to apply. For accounting to work correctly, set this directive to On.
- `RadiusRetransmissionInterval`—Time (in milliseconds) between retransmissions to the RADIUS server. The default is 2000.
- `RadiusRetransmissionCount`—Number of times to retransmit the current RADIUS request before deciding that a given RADIUS server is unreachable and trying the other RADIUS server. The default is 2.
- `RadiusRetransmissionAfterFailure`—Number of times to try the entire procedure for sending a RADIUS request if all attempts to send the previous request failed. The default is 0.
- `RadiusRetryTime`—Time (in seconds) before retrying the primary RADIUS server if it is out of service, or before trying to send requests to any RADIUS server. The default is 300 (5 minutes).

## Configuration Tips

Some general accounting scenarios, and the corresponding configurations, are as follows:

- To only bill for completed calls, set `AccountingServerSide` On, and set `AccountingClientSide` and `AccountingUnsuccessful` Off.
- To bill for features, such as forking, turn `AccountingClientSide` On.
- To gather statistics, measure quality of service, or detect potential network problems, turn `AccountingServerSide`, `AccountingClientSide`, and `AccountingUnsuccessful` On.

# Configurations Example and Call Flows with Debugging Enabled

The following sections illustrate various call scenarios and the associated SIP and RADIUS messaging for each scenario:

- [Successful Call, Server-Side Accounting Enabled, page 18](#)
- [Unsuccessful Call, Server-Side Accounting Enabled, Unsuccessful Accounting Enabled, page 20](#)
- [Forked Call; Server- and Client-Side Accounting Enabled; Unsuccessful Accounting Enabled, page 21](#)

The components involved in the scenarios are listed in [Table 3](#):

**Table 3** *Call-Scenario Components*

Component	Name	IP Address	Hostname
Cisco SIP Proxy Server	Cisco SPS	b.19.174.239	vvs-vitra
FreeRADIUS Server	FreeRADIUS	b.19.174.128	n/a
IP Phone (Cisco 7905)	Cisco 7905	a.22.76.60	n/a
IP Phone (Cisco 7906)	Cisco 7906	a.2.3.4	n/a
IP Phone (Cisco 7907)	Cisco 7907	b.23.252.22	n/a
IP Phone (Cisco 7961)	Cisco 7961	a.22.77.248	n/a

## Successful Call, Server-Side Accounting Enabled

This scenario illustrates a successful call between two registered users, Cisco 7961 and Cisco 7905, through their proxy Cisco SPS. In this scenario, Cisco SPS has server-side accounting enabled for a single RADIUS server and sends Start and Stop records for the call to the RADIUS server. [Table 4](#) provides a summary of the required modifications to the default directives (organized by relevant configuration option tab) to support this RADIUS-based scenario. Please refer to the “[Cisco SPS Interface Reference Appendix](#)” for screen examples showing configuration options for RADIUS-implementation related directives.

**Table 4** *Cisco SPS Directives Modifications for RADIUS-based Implementation for the Successful Call Scenario*

Cisco SPS Configuration Option Screen	Cisco SPS Directive	Default Value	Modified Value
SIP Server Core Configuration	AddRecordRoute	Off	On
Accounting	Accounting	Off	On
	Unsuccessful	Off	On
	PrimaryRadiusServer	Not set	Set to IP, port, and secret of the Free RADIUS server

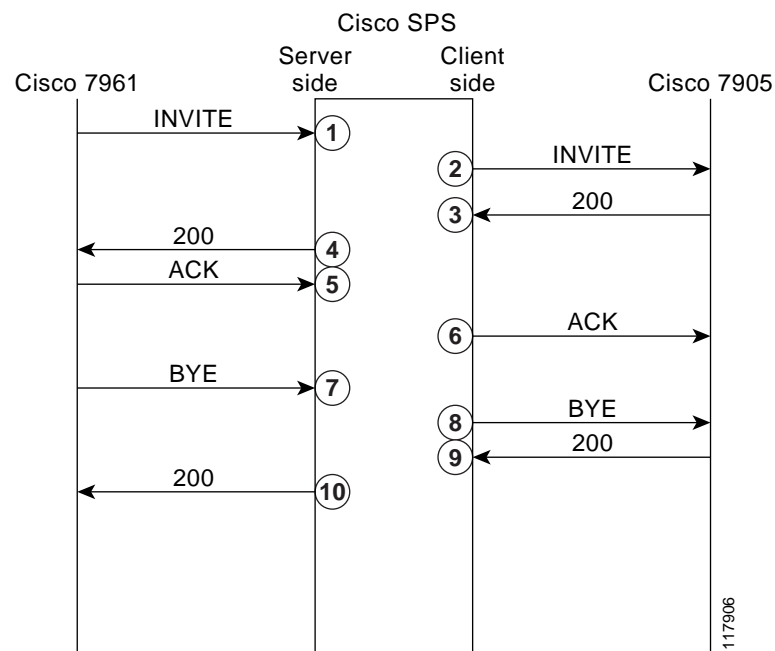
**Table 4** Cisco SPS Directives Modifications for RADIUS-based Implementation for the Successful Call Scenario

Cisco SPS Configuration Option Screen	Cisco SPS Directive	Default Value	Modified Value
Debugs and Logs	StateMachine	Off	On
	Radius	Off	On
	LogLevel	warn	debug

## Call Flow

The call flow for this scenario as shown in [Figure 4](#).

**Figure 4** Basic Successful Call



The following list summarizes the call process outlined in [Figure 4](#):

1. Cisco 7961 wants to call Cisco 7905. Cisco 7961 sends INVITE for Cisco 7905 to Cisco SPS.
2. Cisco SPS forwards the INVITE to Cisco 7905.
3. Cisco 7905 answers the call and returns 200.
4. Cisco SPS forwards 200 to Cisco 7961 and sends **server side Start**.
5. Cisco 7961 sends the ACK to Cisco SPS.
6. Cisco SPS forwards the ACK to Cisco 7905.
7. Cisco 7961 sends BYE to Cisco SPS.
8. Cisco SPS forwards the BYE to Cisco 7905.
9. Cisco SPS receives the 200 for the BYE from Cisco 7905.

10. Cisco SPS forwards the 200 for the BYE to Cisco 7961 and sends server side successful Stop.

**Note**

---

For complete listings of records, debug output and error logs related to the RADIUS interface Cisco SPS implementation, please refer to [“RADIUS Interface for Cisco SPS Scenario Output Appendix”](#).

---

## Unsuccessful Call, Server-Side Accounting Enabled, Unsuccessful Accounting Enabled

This scenario illustrates an unsuccessful call from a registered user (Cisco 7961) to an unregistered and unknown user (in this scenario named Cisco 7000) through the Cisco SPS (serving as a proxy for Cisco 7961). The Cisco SPS has server-side accounting enabled and sends a Stop record for the unsuccessful call attempt to the FreeRADIUS server.

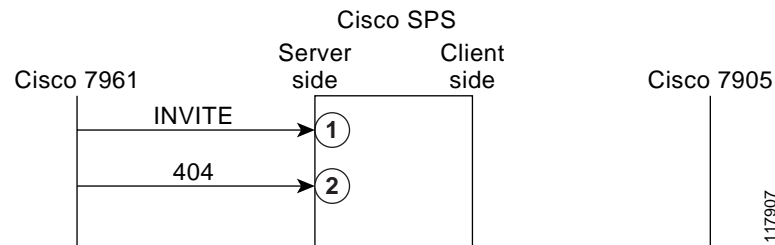
### Configuration

Cisco SPS is configured the same as described in the [“Successful Call, Server-Side Accounting Enabled”](#) section on page 18.

## Call Flow

Figure 5 illustrates an unsuccessful call flow.

Figure 5 Basic Unsuccessful Call



The process illustrated in Figure 5 is as follows:

1. Cisco 7961 wants to call (unregistered user) Cisco 7000. Cisco 7961 sends an INVITE for Cisco 7000 to Cisco SPS.
2. 7000 is not a registered user, so Cisco SPS returns 404 and sends server side unsuccessful Stop.



Note

For complete listings of records, debug output and error logs related to the RADIUS interface Cisco SPS implementation, please refer to [“RADIUS Interface for Cisco SPS Scenario Output Appendix”](#).

## Forked Call; Server- and Client-Side Accounting Enabled; Unsuccessful Accounting Enabled

This scenario illustrates a successful call between two registered users, Cisco 7961 and Cisco 7905, through their proxy, Cisco SPS. Cisco 7905 is registered as both Cisco 7905 and Cisco 7906. Cisco 7906 does not respond, resulting in a 408 (time-out). Cisco 7905 responds with 486 (busy), resulting in Call Forward Busy to Cisco 7907. Cisco 7907 answers, resulting in a successful call.

## Configuration

Cisco SPS is configured with server- and client-side accounting enabled to a single RADIUS server, as well as with call forward busy enabled for Cisco 7905 to Cisco 7907. Unsuccessful accounting is enabled as well. In order to facilitate the correlation of the various server- and client side Accounting-Request packets, Cisco SPS is configured to include the To, From, and CSeq SIP headers in the Accounting-Request packets. This is accomplished by modifying the default configuration with which Cisco SPS was first installed as described in this section. Table 5 provides a summary of the required modifications (organized by relevant configuration option tab or screen) to support this RADIUS-based scenario. Please refer to the [“Cisco SPS Interface Reference Appendix”](#) for screen examples illustrating configuration options for specifying RADIUS-implementation related options.

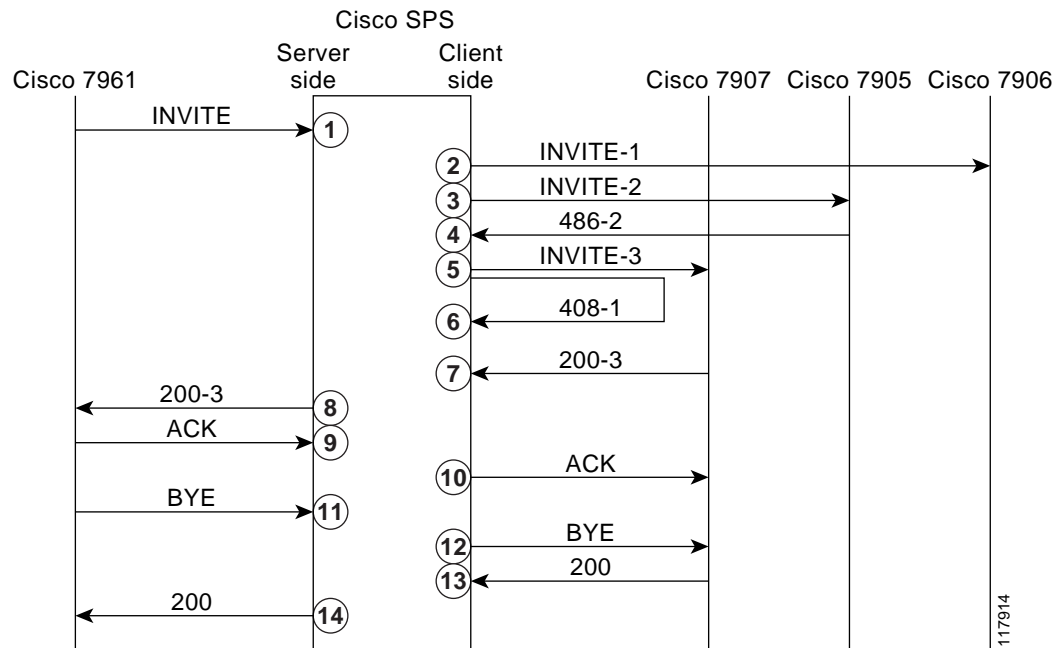
**Table 5** Cisco SPS Modifications for RADIUS-based Implementation for Forked-Call Scenario

Cisco SPS Configuration Option Screen	Cisco SPS Directive	Default Value	Modified Value
SIP Server Core Configuration	AddRecordRoute	Off	On
Accounting	Accounting	Off	On
	Unsuccessful	Off	On
	PrimaryRadiusServer	Not set	Set to IP, port, and secret of the Free RADIUS server
	SIP Headers	Not set	Added from, to, CSeq
Call Forward	Busy	Off	On
	Diversion Header Name	CC-Diversion	Diversion (This is an optional change.)
Subscribers	Not applicable		Added a new subscriber, Cisco 7905, with call forward busy (CFB) to Cisco 7907
Debug and Logs	State Machine	Off	On
	Radius	Off	On
	LogLevel	warn	debug

## Call Flow

The call flow in [Figure 6](#) is the same as the one shown in [Figure 4](#), except that the endpoints have changed.

Figure 6 Forked Call-Flow77



The following process stages summarize the protocol illustrated in Figure 6.

1. Cisco 7961 wants to call Cisco 7905. Cisco 7961 sends INVITE for Cisco 7905 to Cisco SPS.
2. Cisco SPS forks the INVITE and INVITE is sent to Cisco 7905 (registration is implied).
3. Cisco SPS forks the INVITE and INVITE is sent to Cisco 7906 (registration is implied).
4. Cisco 7905 is busy and returns 486. Cisco SPS sends client side unsuccessful Stop.
5. Cisco 7905 has a Call Forward Busy contact set to Cisco 7907, so Cisco SPS forwards INVITE to Cisco 7907.
6. INVITE to Cisco 7906 times out. Cisco SPS generates internal 408 and sends client side unsuccessful Stop.
7. Cisco 7907 answers the call, returning 200. Cisco SPS sends client side Start.
8. Cisco SPS forwards 200 to Cisco 7961 and sends server side Start.
9. Cisco 7961 sends ACK.
10. Cisco SPS forwards the ACK to Cisco 7907.
11. Cisco 7961 sends BYE to Cisco SPS.
12. Cisco SPS forwards the BYE to Cisco 7907.
13. Cisco SPS receives the 200 for the BYE from Cisco 7907 and sends client side successful Stop.
14. Cisco SPS forwards the 200 for the BYE to Cisco 7961 and sends server side successful Stop.



**Note**

For complete listings of records, debug output and error logs related to the RADIUS interface Cisco SPS implementation, please refer to “[RADIUS Interface for Cisco SPS Scenario Output Appendix](#)”.

