Cisco BroadWorks

SIP Network Interface

Interworking Guide

Release 23.0

Document Version 2

Notification

The BroadSoft BroadWorks has been renamed to Cisco BroadWorks. Beginning in September 2018, you will begin to see the Cisco name and company logo, along with the new product name on the software, documentation, and packaging. During this transition process, you may see both BroadSoft and Cisco brands and former product names. These products meet the same high standards and quality that both BroadSoft and Cisco are known for in the industry.

Copyright Notice

Trademarks

Document Revision History

Release	Version	Reason for Change	Date	Author
8	1.1	Introduced Release 8 changes.	July 22, 2002	Sam Hoffpauir
8	1.2	Added section for interface changes between Release 7 and 8.	August 19, 2002	Sam Hoffpauir
8	1.3	Document revision management. No changes to document.	August 19, 2002	Nora Navin
8	1.4	Fixed typographical errors related to Remote-Party-ID in section 6.5.	September 15, 2002	Sam Hoffpauir
9	2.1	Updated with Release 9 interface changes.	February 18, 2003	Sam Hoffpauir
9	2.4	Updated with RFC 3261 loose routing support.	June 25, 2003	Sam Hoffpauir
10	1	Updated with Release 10 interface changes.	December 3, 2003	Sam Hoffpauir
11	1	Updated with Release 11 interface changes.	April 20, 2004	Sam Hoffpauir
12	1	Updated with Release 12 interface changes.	November 30, 2004	Sam Hoffpauir
12	2	Added video codec requirements.	January 10, 2004	Sam Hoffpauir
12	3	Added clarifications and made editorial fixes.	April 13, 2005	Sam Hoffpauir
12	4	Added clarifications and made editorial fixes.	April 27, 2005	Sam Hoffpauir
13	1	Updated with Release 13 interface changes.	September 9, 2005	Sam Hoffpauir
14.0	1	Updated document for re-branding.	February 14, 2006	Roberta Boyle
14.0	1	Updated document with Release 14 Interface changes.	September 5, 2006	Sam Hoffpauir
14.0	1	Edited document.	September 26, 2006	Patricia Renaud
14.0	2	Updated document with fax information.	October 25, 2006	Stephane Bastien
14.0	2	Edited changes.	October 27, 2006	Stephane Bastien
14.0	3	Corrected section 3.12 Offer/Answer and Early Media Support.	December 12, 2006	Martin Perron
14.0	3	Edited changes.	January 3, 2007	Patricia Renaud
14.0	4	Updated document with Release 14.sp1 and Release 14.sp2 content.	August 27, 2007	Martin Piotte
14.0	4	Edited changes and published document.	September 28, 2007	Andrea Fitzwilliam
14.0	5	Updated document with Release 14.sp3 and Release 14.sp4 content.	December 20, 2007	Martin Piotte
14.0	5	Updated section *3.10 Diversion Indication in SIP (RFC 5806)* for EV 57438.	February 5, 2008	Roberta Boyle

Release	Version	Reason for Change	Date	Author
14.0	5	Edited changes and published document.	April 7, 2008	Andrea Fitzwilliam
14.0	6	Removed incorrect information.	May 9, 2008	Martin Piotte
14.0	6	Edited changes and published document.	May 13, 2008	Andrea Fitzwilliam
14.0	7	Updated document with Release 14.sp5, Release 14.sp6, and Release 14.sp7 content.	June 13, 2008	Martin Piotte
15.0	1	Updated document for Release 15.0.	June 17, 2008	Martin Piotte
15.0	1	Edited changes and published document.	August 11, 2008	Andrea Fitzwilliam
15.0	2	Added description of the *Via* header and AA loop detection.	January 21, 2008	Martin Piotte
16.0	1	Updated document for Release 16.0.	May 19, 2009	Martin Piotte
16.0	1	Updated list of Cisco BroadWorks proprietary parameters for the *Diversion* and *History-Info* headers.	June 22, 2009	Martin Piotte
16.0	1	Updated section *3.10 Diversion Indication in SIP (RFC 5806)* for EV 89837.	June 24, 2009	Roberta Boyle
16.0	1	Edited changes, which include those for EVs 95196 and 89837, and published document.	June 30, 2009	Andrea Fitzwilliam
16.0	2	Updated provisional response section to add limitations, for EV 104113 and EV 104164.	February 16, 2010	Eric Bernier
16.0	2	Edited changes and published document.	February 23, 2010	Andrea Fitzwilliam
17.0	1	Updated document for Release 17.0.	April 8, 2009	Martin Piotte
17.0	1	Edited changes and published document.	April 21, 2010	Margot Hovey-Ritter
17.0	2	Added information about SIP timers to section *3.1.3 SIP Timers*.	October 15, 2010	Martin Piotte
17.0	2	Edited changes and published document.	October 20, 2010	Andrea Fitzwilliam
17.0	3	Added section *3.47 Calling Line ID Unavailable and Anonymous*. Updated section *3.7 Calling Party's Category tel URI Parameter (draft-mahy-iptel-cpc-00)/isup-oli Parameter Support* with CPC in GTD and configurable CPC values.	October 20, 2010	Martin Piotte
17.0	3	Edited changes and published document.	November 5, 2010	Andrea Fitzwilliam
17.0	4	Updated section *3.47 Calling Line ID Unavailable and Anonymous*. Added new rules for restricted identity based on EV 126844.	February 23, 2011	Eric Bernier

Release	Version	Reason for Change	Date	Author
14.0	4	Updated BroadSoft address and references to Xchange.	March 25, 2011	Goska Auerbach
17.0	4	Edited changes and published document.	June 22, 2011	Jessica Boyle
18.0	1	Updated document for Release 18.0.	November 4, 2011	Martin Piotte
18.0	1	Edited changes and published document.	November 8, 2011	Patricia Renaud
19.0	1	Updated document for Release 19.0 features.	September 18, 2012	Doug Sauder
19.0	1	Edited changes and published document.	October 24, 2012	Patricia Renaud
19.0	2	Removed appendix entitled *SIP Protocol Requirements for Cisco BroadWorks Features* for EV 177212.	March 27, 2013	Doug Sauder
19.0	2	Edited changes and published document.	May 28, 2013	Jessica Boyle
19.0	3	Updated section *3.43 Transparent Proxying of SIP Headers and Options* for EV 190819.	July 5, 2013	Doug Sauder
19.0	3	Added additional information about configurable treatments and the *Reason* header for EV 196816.	July 25, 2013	Doug Sauder
20.0	1	Updated document for Release 20.0 features.	September 10, 2013	Doug Sauder
20.0	1	Edited changes and published document.	October 29, 2013	Joan Renaud
20.0	2	Added information about the *P-BroadWorks-Endpoint-Owner-ID* header.	December 18, 2013	Doug Sauder
20.0	2	Edited changes and published document.	April 15, 2014	Joan Renaud
21.0	1	Updated document for Release 21.0 features.	November 3, 2014	Doug Sauder
21.0	1	Updated the legal notice and edited changes.	November 28, 2014	Joan Renaud
21.0	1	Rebranded and published document.	December 19, 2014	Joan Renaud
21.0	2	Added information about how Cisco BroadWorks applies “history” privacy to the *Diversion* header.	February 27, 2015	Doug Sauder
21.0	2	Added rebranded server logos, edited changes, and published document.	March 9, 2015	Joan Renaud
21.0	3	Changed *networkSendHistoryInfo* to *useHistoryInfoOnNetworkSide* for PR-49406.	January 29, 2016	Doug Sauder
21.0	3	Edited changes and published document.	June 7, 2016	Joan Renaud
22.0	1	Updated document for Release 22.0 features.	September 23, 2016	Doug Sauder

Release	Version	Reason for Change	Date	Author
22.0	1	Edited changes and published document.	December 9, 2016	Joan Renaud
22.0	2	Corrected information about error responses to the UPDATE request for PR-57393.	July 19, 2018	Doug Sauder
23.0	1	Updated document for Release 23.0 features.	October 20, 2018	Doug Sauder
23.0	1	Edited changes and published document.	November 5, 2018	Patricia Renaud
23.0	2	Rebranded product name for Cisco and published document.	March 20, 2019	Joan Renaud

Table of Contents

Table of Figures

1 Summary of Changes

This section describes the changes to this document for each release and document version.

1.1 Changes for Release 23.0, Document Version 2

This version of the document includes the following change:

- Rebranded product name for Cisco.

1.2 Changes for Release 23.0, Document Version 1

The following SIP access interface changes were made to this document for Release 23.0. They are the interface differences between Cisco BroadWorks Release 22.0 and Release 23.0.

- A new SIP system parameter *supportHeaderLevelPrivacy* controls whether Cisco BroadWorks should apply anonymous presentation for calling user's identity in response to receiving the "header" value.
- The SIP system parameters that control forking support are changed to support a dynamic switch from multiple dialog mode to single dialog mode.
- The SIP system parameter *supportNoForkOption* controls whether Cisco BroadWorks supports the "no-fork" directive in the *Request-Disposition* header.
- The SIP system parameter *support199* controls whether Cisco BroadWorks supports the 199 (Dialog Terminated) provisional response.
- The SIP system parameter *proxyForkingProvisionalResponses* controls whether Cisco BroadWorks should relay provisional responses from secondary device endpoints such as Shared Call Appearance device endpoints.
- Cisco BroadWorks ignores the *RFC 3398* policy if *P-Early-Media* support is enabled and the terminating device sends a *P-Early-Media* header.
- The SIP system parameter *suppressRFC3312Preconditions* can take the value "suppressIfSingleDialog", which causes Cisco BroadWorks to suppress preconditions if the originating session operates in single-dialog mode.
- Cisco BroadWorks supports preconditions negotiation when it terminates to the Media Server.

1.3 Changes for Release 22.0, Document Version 2

- Corrected information about error responses to the UPDATE request for PR-57393.

1.4 Changes for Release 22.0, Document Version 1

The following SIP access interface changes were made to this document for Release 22.0. They are the interface differences between Cisco BroadWorks Release 21.0 and Release 22.0.

- Depending on configuration, Cisco BroadWorks may send the "header" value in the *Privacy* header.
- Depending on configuration, Cisco BroadWorks may support the *sip.video* media feature tag in accordance with *RFC 3840* and GSMA IR.94.

- Depending on configuration, Cisco BroadWorks can suppress preconditions attributes.
- Cisco BroadWorks supports a new configuration option for forking support. The SIP system parameter *networkForkingSupport* has a new option "singleDialogWIthUPDATEIfAllowed", which prevents Cisco BroadWorks from sending an UPDATE request to an endpoint that does not support it.
- In order to facilitate services, Cisco BroadWorks may add the new proprietary parameter *redir-mobile* to the *X-BroadWorks-DNC* header.
- To facilitate services, Cisco BroadWorks can add new proprietary parameters *x-bw-phone-list-name* and *x-bw-igc* to the *History-Info* and *Diversion* headers.
- Cisco BroadWorks uses the info package *x-broadworks-transfer-notification* to provide call transfer notification between two Application Servers. Cisco BroadWorks uses this info package only for a very specific call transfer scenario.

The following changes, which are not related to changes in Release 22.0, were made in this document version:

- Added a detailed explanation about how Cisco BroadWorks performs DNS queries for NAPTR, SRV, AAAA, and A records.

1.5 Changes for Release 21.0, Document Version 3

The following change was made in this document version:

- Changed *networkSendHistoryInfo* (old name) to *useHistoryInfoOnNetworkSide* (new name) in section *3.10.2.3 History-Info and Diversion Interworking (RFC 6044)* for PR-49406.

1.6 Changes for Release 21.0, Document Version 2

The following changes were made in this document version:

- Added information about how Cisco BroadWorks applies "history" privacy to the *Diversion* header.
- Added rebranded server icons.

1.7 Changes for Release 21.0, Document Version 1

The following SIP access interface changes were made to this document for Release 21.0. They are the interface differences between Cisco BroadWorks Release 20.0 and Release 21.0.

- Added information about the potential problems of a quick re-INVITE and the related configuration.
- Added information about how Cisco BroadWorks builds the *Call-ID* header value and the related configuration.
- Added information about how Cisco BroadWorks builds the *branch* parameter of the *Via* header and the related configuration.
- Added information about support for the *cause* URI parameter as recommended in *RFC 4458*.
- Added information about *History-Info* header and *Diversion* header interworking, following the recommendations of *RFC 6044*.
- Added new parameter values for the *X-BroadWorks-DGC* header.

The following changes, which are not related to changes in Release 21.0, were made in this document version:

- Corrected outdated information about system parameter configuration for multiple dialog support for EV 220679.
- Added a table of all SIP system parameters as an appendix.

1.8 Changes for Release 20.0, Document Version 2

The following change was made in this document version:

- Added information about the proprietary *P-BroadWorks-Endpoint-Owner-ID* header.

1.9 Changes for Release 20.0, Document Version 1

The following SIP network interface changes were made to this document for Release 20.0. They are the differences in the interface between Cisco BroadWorks Release 19.0 and Release 20.0.

- Added *external-vm-deposit* as a new parameter in the *Diversion* header.
- Added these new values for the *Reason* header: "sac-group-rejection", "sac-group-orig-rejection", "sac-group-term-rejection", and "ea-call-push".
- Added an explanation of Cisco BroadWorks new ability to inject an unrecognized header (such as *User-to-User*) from a 302 response or REFER request into an outgoing INVITE request.
- Added the new *X-BroadWorks-DGC*, *X-BroadWorks-DNC*, and *X-BroadWorks-Correlation-Info* headers.
- Added information about the ability of the secondary Application Server to act as a stateless proxy.

The following additional changes were made to the document:

- Removed information about the Call Recording interface, since this information was duplicated from the *Cisco BroadWorks Call Recording Guide*.
- Added the full Augmented Backus-Naur Format (ABNF) syntax definition for the *X-BroadWorks-DGC* and *X-BroadWorks-DNC* headers.

1.10 Changes for Release 19.0, Document Version 3

The following changes were made in this document version:

- Updated section *3.43 Transparent Proxying of SIP Headers and Options* to explain that Cisco BroadWorks may transparently proxy the *Accept-Contact* header when patch *AP.as.19.0.574.ap189579* (or *AP.as.19.sp1.574.ap189579*) is applied.
- Added information about the *P-Access-Network-Info* header.
- Added information about SIP headers in RFC 3455 that were not mentioned in earlier document versions.
- Added additional information about configurable treatments and the *Reason* header.

1.11 Changes for Release 19.0, Document Version 2

The following change was made in this document version:

- Removed appendix entitled *SIP Protocol Requirements for BroadWorks Features* for EV 177212.

1.12 Changes for Release 19.0, Document Version 1

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 19.0. They are the differences in the interface between Cisco BroadWorks Release 18.0 and Release 19.0.

- Added new diversion reasons for Find Me/Follow Me service and Charge Number service.
- Updated to indicate that the P-Charge-Info header may contain a sip URI or tel URI.
- Updated the Cisco BroadWorks Call Recording Metadata to version 2.0.
- Added information about Cisco BroadWorks IPv4/IPv6 dual stack support.

1.13 Changes for Release 18.0, Document Version 1

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 18.0. They are the interface differences between Cisco BroadWorks Release 17.0 and Release 18.0.

- Added configurable transport in the *Contact* header (section *3.1.2.1 Differences between UDP and TCP Transports for SIP*).
- Added Application Server configured Originating Trunk Group support (section *3.8 Incoming/Outgoing OTG Support*).
- Added Destination Trunk Group (DTG) for terminating calls and modified DTG syntax (section *3.9 Destination Trunk Group Support*).
- Updated Short Message Service (SMS) and SIP MESSAGE support (section *3.26 SIP Extension for Instant Messaging (RFC 3428)*).
- Added support for *max-fs* and *max-mbps* H.264 SDP payload format options in the SDP (section *3.33.2.4 SDP Handling – Video Streaming Enabled on Media Server*).
- Updated Connected Line Identification Presentation (COLP), which is now supported in the UPDATE and re-INVITE messages (section *3.37 Connected Line Identification Presentation*).
- Added Call Recording (section *3.48 Call Recording*).
- Added *Priority*, *Resource-Priority*, and *P-DCS-OSPS* SIP headers for emergency calls (section *3.49 Priority, Resource-Priority, and P-DCS-OSPS SIP Headers for Emergency Calls*).
- Added support for IPv6 (section *3.50 IPv6 and* IPv4/IPv6 Dual Stack Support).

1.14 Changes for Release 17.0, Document Version 4

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 17.0. They are the interface differences between Cisco BroadWorks Release 17.0 version 3 and Release 17.0 version 4.

- Updated section *3.47 Calling Line ID Unavailable and Anonymous* for EV 126844. Cisco BroadWorks has been enhanced to support additional keywords in order to treat a call with calling line identity restriction.

1.15 Changes for Release 17.0, Document Version 3

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 17.0. They are the interface differences between Cisco BroadWorks Release 17.0 version 2 and Release 17.0 version 3.

- Added section *3.47 Calling Line ID Unavailable and Anonymous*.
- Updated section *3.7 Calling Party's Category tel URI Parameter (draft-mahy-iptel-cpc-00)/isup-oli Parameter Support* with CPC in GTD and configurable CPC values.

1.16 Changes for Release 17.0, Document Version 2

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 17.0. They are the interface differences between Cisco BroadWorks Release 17.0 version 1 and Release 17.0 version 2.

- Added SIP timer information to section *3.1.3 SIP Timers*.

1.17 Changes for Release 17.0, Document Version 1

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 17.0. They are the interface differences between Cisco BroadWorks Release 16.0 and Release 17.0.

- Addition of section *3.32.2 Reason Header on Redirection*.
- Addition of section *3.42.2 Legacy Automatic Callback*.
- Addition of section *3.46 P-Camel Headers*.
- Addition of an optional classmark in the *From* header.
- Addition of the route-point reason and intercept exempt parameter in the *Diversion* and *History-Info* headers.

1.18 Changes for Release 16.0, Document Version 2

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 16.0. They are the interface differences between Cisco BroadWorks Release 16.0 version 1 and Release 16.0 version 2.

- Updated section *0* .

1.19 Changes for Release 16.0, Document Version 1

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 16.0. They are the interface differences between Cisco BroadWorks Release 15.sp2 and Release 16.0.

- Introduction of the *X-BroadWorks-DNC* header.
- Introduction of the *P-Charge-Info* header.
- Transparent proxy of unknown *SIP* headers and options.
- Advice of Charge support.
- Enhancements to media changes between early and established sessions.
- Replaced *maxHops* values with *defaultMaxRedirectionDepth* in section *3.10 Diversion Indication in SIP (RFC 5806)* for EV 89837.
- Made changes for EV 95196.

1.20 Changes for Release 15.sp2

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 15.sp2. They are the interface differences between Cisco BroadWorks Release 15.0 and Release 15.sp2.

- Addition of *answered-count* parameter to the *Diversion* and *History-Info* header for post-answer loop detection.

1.21 Changes for Release 15.0

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 15.0. They are the interface differences between Cisco BroadWorks Release 14.sp6 and Release 15.0.

- Codec details in Cisco BroadWorks Video IVR Support.
- Use of the *network-inhibited* parameter in the *Diversion* header.
- Call Completion Services.

1.22 Changes for Release 14.sp7

The following SIP network interface changes were made to this document for Release 14.sp7. They are the interface differences between Cisco BroadWorks Release 14.sp6 and Release 14.sp7.

- Support of the *History-Info* header.

1.23 Changes for Release 14.sp6

The following SIP network interface changes were made to this document for Release 14.sp6. They are the interface differences between Cisco BroadWorks Release 14.sp5 and Release 14.sp6.

- Support for the *P-Called-Party-ID* SIP header.

1.24 Changes for Release 14.sp5

The following SIP network interface changes were made to this document for Release 14.sp5. They are the interface differences between Cisco BroadWorks Release 14.sp4 and Release 14.sp5.

- Support for the *AccessCode* SIP header.

1.25 Changes for Release 14.sp4

There were no changes to this document for Release 14.sp4.

1.26 Changes for Release 14.sp3

The following SIP network interface changes were made to this document for Release 14.sp3. They are the interface differences between Cisco BroadWorks Release 14.sp2 and Release 14.sp3.

- Support for the *Retry-After* header for congestion control.
- Support for a Destination Trunk Group indication.
- Improvements to the Session Timers in SIP.
- Configurability of the Forking Proxy policy for media changes.

- Support for Connected Line Identification Presentation (COLP).
- Support for External Custom Ringback.

1.27 Changes for Release 14.sp2

The following SIP network interface changes were made to this document for Release 14.sp2. They are the interface differences between Cisco BroadWorks Release 14.sp1 and Release 14.sp2.

- Support for configurable treatments allows configurability of the SIP status code mapping and system treatments mapping. It also provides support for the *Reason* header, as defined in *RFC 3326*, which includes support of Q.850 cause codes.

1.28 Changes for Release 14.sp1

The following SIP network interface changes were made to this document for Release 14.sp1. They are the interface differences between Cisco BroadWorks Release 14.0 and Release 14.sp1.

- Support for signaling in the *Diversion* header a "diversion inhibited" condition. This applies when Cisco BroadWorks sends an INVITE message for which diversion is inhibited, either because of FAC dialing or implicitly for Hunt Group and Call Center redirection.
- Enhancement to the Message Summary (optionally) to send the number of saved and urgent messages in addition to new messages.
- Support for sending 503 Service Unavailable responses (optionally) during overload condition. 503 responses can be selected as an alternative to sending 302 Moved Temporarily responses, or ignoring the request.
- Support for proxying the *P-Early-Media* header (draft-ejzak-sipping-p-em-auth-02).
- Improved support for proxying message bodies though INFO, ACK, PRACK (and responses), and UPDATE (and responses).
- Support for Dual-Tone Multi-Frequency (DTMF) signaling in INFO messages.

1.29 Changes for Release 14.0

The following SIP network interface changes were made to this document for Release 14.0. They are the interface differences between Cisco BroadWorks Release 13.0 and Release 14.0.

- Support of the *Charge* header to facilitate interworking with GR.394-compliant ISUP networks, allowing Cisco BroadWorks to provision and populate the charging information independently of the subscriber's calling line identity.
  - When the Charge Number service is assigned, the configured charge number for the user is included in the call detail records (CDRs) generated for the user's originating calls and in the *Charge* header of the SIP INVITEs for calls originated by the user.
- Support for the E911 i2 recommendation by the National Emergency Number Association (NENA) to enhance the Cisco BroadWorks E911 offering.
  - This enhancement to Cisco BroadWorks implements a subset of the recommendations for the V5 interface defined in the National Emergency Number Association (NENA) 911 i2 draft document. The V5 interface defines the protocol between the call server (Cisco BroadWorks) and the 911 redirect server.

  - During an emergency call, when a SIP redirect response (SIP 3XX) is received from a redirect server, the generated INVITE from the redirect response populates the *Request-URI, FROM, TO*, and *P-Asserted-Identity* headers as recommended in the NENA i2 specification. This allows the Emergency Services Gateway (ESGW) to route the call to the appropriate E911 Selective Router (SR) and then have the call routed to the appropriate Public Safety Answering Point (PSAP).

- TCP Connection Management enhancements, including explicit transport configuration. Cisco BroadWorks recommends devices to use connection reuse when using connection-oriented protocols such as Transmission Control Protocol (TCP).
- Full support of the UPDATE method including early media offer/answer exchanges.
- Full support of *RFC 3262* Reliability of Provisional Response including offer/answer exchanges of Early Media Support.
- Full support of *RFC 3959/3960* Early Sessions.
- Support of forking capability to comply with *RFC 3261* offer/answer exchanges.
- Support of flexible 2xx response handling to allow interoperability flexibility with devices of varying degrees of compliance to the offer/answer exchanges and early media support.
- Support for inhibiting call redirections on remote call control platforms. Cisco BroadWorks now supports the ability to insert a diversion entry with a counter that exceeds the maximum allowed redirections in the network for a call causing remote call control platforms such as legacy Class 5 switches, to disable Redirection services on the switch and terminate the call directly to the subscriber's phone. This capability is useful in providing services to legacy remote call control platform subscribers without requiring a second Public Switched Telephone Network (PSTN) phone number.
- Support of the H.264 video codec. Cisco BroadWorks now supports the H.264 video codec in addition to the H.263-1998 and H.263-2000 video codecs previously supported.
- Integrated support of T.38 fax. Cisco BroadWorks supports both sending and receiving T.38 fax.

1.30 Changes for Release 13.0

The following SIP network interface changes were made to this document for Release 13.0. They are the interface differences between Cisco BroadWorks Release 12.0 and Release 13.0.

- Support of *RFC 3398* early media transitions to allow receipt of a 180 Ringing without Session Description Protocol (SDP) for a CPG received by the network device, after receiving a 183 Session Progress with SDP associated with the Audio Compression Manager (ACM) received by the network device. Prior to this behavior, Cisco BroadWorks inserted ringback via the Media Server causing garbled ringback.

1.31 Changes for Release 12.0

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 12.0. They are the interface differences between Cisco BroadWorks Release 11.0 and Release 12.0.

- Addition of extended diversion reasons (for example, call-center, hunt-group, and so on).
- Use of SIP contact advancing for Application Server overload conditions. Upon overload conditions detected by the Cisco BroadWorks Application Server, Cisco BroadWorks may send a *302 Moved Temporarily* response to the device to force it to the secondary Application Server.
- Video device requirements. Cisco BroadWorks supports integrated video devices and video-only devices for the BroadWorks Video Add-On service. Additionally, Cisco BroadWorks supports Video IVR services. Device requirements are added for video device interoperability with Cisco BroadWorks, and clarifications are added for Cisco BroadWorks video codec support.
- Configurable support of content types. Cisco BroadWorks is enhanced to allow configurability of the content types accepted by the Application Server.
- Equal Access Enhancements. Cisco BroadWorks is enhanced to allow Preferred Inter-exchange Carriers (PICs) provisioning on a per-user basis. Cisco BroadWorks also supports the ability to provision PICs on groups and enterprises (for enterprises, on a per-country code basis). Call processing PIC selection precedence for outgoing call setup messages is as follows: user, group, enterprise. Thus, if a user has a CIC, then it overrides the CIC settings of the group and enterprise.
- Support of TCP. Cisco BroadWorks now provides TCP support in addition to User Datagram Protocol (UDP) support.
- Support of 3GPP IP Multimedia Subsystem (IMS) interface. Cisco BroadWorks supports the 3GPP IMS interface, allowing the Cisco BroadWorks Application Server to communicate with 3GPP-compliant Call Session Control Function (CSCF) and Application Server components.

1.32 Changes for Release 11.0

The following Session Initiation Protocol (SIP) network interface changes were made to this document for Release 11.0. They are the interface differences between Cisco BroadWorks Release 10.0 and Release 11.0.

- Support of *RFC 3311*, the SIP UPDATE method. This support includes the ability to send and receive the SIP UPDATE request for unconfirmed dialog exchanges. Confirmed dialogs should continue to use the re-INVITE mechanism to alter session/dialog information.
- Enhanced SDP management support. This support provides the ability for Cisco BroadWorks to be fully compliant to *RFC 3264* in addition to enabling enhanced services. This support includes the following changes:
  - An SDP processed by Cisco BroadWorks is “branded” as it passes through, such that the devices exchanging SDPs view Cisco BroadWorks as the owner of the SDP. The **v, o, s** lines are changed by Cisco BroadWorks as part of branding an SDP.

- Processing of a “hold” SDP fully supports the *RFC 3264* specification. However, since some devices do not currently support this, Cisco BroadWorks also supports the deprecated way of handling a “hold” SDP. For the various SDP specifications identifying it as a “hold” SDP, see *RFC 3264*. (The deprecated way of identifying a “hold” SDP is to use 0.0.0.0 in the c-line of the SDP.)
- Cisco BroadWorks is now able to handle video applications from an SDP perspective, by supporting multiple media streams (“**m**” lines) in an SDP.

- Support of *FROM* header *tel-URI* parameter and *URI* parameter for calling party category. This support includes support of the calling party category *tel-URI* parameter based on draft-mahy-iptel-cpc-00. It also includes support of the *isup-OLI URI* parameter. Cisco BroadWorks can optionally send out calling party category information in either parameter, based on the network interface configuration. It is up to the softswitch or network gateway to map the contents of this parameter into the appropriate legacy protocol (for example, Integrated Services User Part [ISUP], Primary Rate Interface [PRI], and so on)
- Support of *FROM* header parameter *OTG*, originating trunk group. This support includes support of an *Originating Trunk Group (OTG) URI* parameter, which contains an originating identifier to aid in call routing.

1.33 Changes for Release 10.0

The following SIP network interface changes were made to this document for Release 10.0. They are the interface differences between Cisco BroadWorks Release 9.0 and Release 10.0.

- Addition of privacy support in the *Diversion* header per draft-levy-sip-diversion-06. In draft-levy-sip-diversion-06, the *diversion-privacy* parameter has been added to the diversion-params. The *diversion-privacy* parameter provides four types of privacy indication: no privacy, name privacy, URI privacy, and name and URI privacy (full calling party identity privacy). The values of privacy parameter can be “privacy=off”, “privacy=name”, “privacy=URI”, or “privacy=full” for no privacy, name privacy, URI privacy, and full privacy, respectively. Although Cisco BroadWorks can receive any value in the privacy parameter, Cisco BroadWorks only populates the *diversion-privacy* parameter with a value of “privacy=full”, for diversion entries added by Cisco BroadWorks.
- Sending of privacy headers for all calls instead of just for calls with restricted calling line identity. Cisco BroadWorks now always includes the appropriate privacy headers for the following privacy versions: privacy-00, privacy-03, and *RFC 3323*. The *RFC 3323-Japan Privacy Version* is unaffected by this feature.
- Support for draft-ietf-sip-session-timer-12. Cisco BroadWorks previously supported the draft-ietf-sip-session-timer-04 version of Session Timer. This version was not compatible with subsequent versions. In Release 10.0, Cisco BroadWorks is compliant to draft-ietf-sip-session-timer-12. Cisco BroadWorks only includes the *session-expires* header when required, and always chooses the remote user agent to refresh the dialog when possible, as specified in draft-ietf-sip-session-timer-12.

1.34 Changes for Release 9.0

The following SIP network interface changes were made to this document for Release 9.0. They are the interface differences between Cisco BroadWorks Release 8.0 and Release 9.0.

- Cisco BroadWorks has added the ability to echo back unknown *Via header* parameters, such as *Via: SIP/2.0/UDP host;nrtag=0.000224.24.000*. Prior to Release 9.0, Cisco BroadWorks would remove unknown *Via header* parameters rather than echoing the unknown parameters.
- Cisco BroadWorks has added support for an additional privacy mechanism via RFC 3323, A Privacy Mechanism for the Session Initiation Protocol (SIP), and RFC 3325, Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks.
- Cisco BroadWorks has added full support of RFC 3265, Session Initiation Protocol (SIP)-Specific Event Notification.
- Cisco BroadWorks has added support for a variety of event packages that make use of *RFC 3265*, including packages that obtain the calling name (CNAM) for a call and allow receipt and delivery of third-party voice message waiting indicators to subscribers using third-party Voice Mail systems.
- Cisco BroadWorks has added support for loose routing in compliance with *RFC 3261*.

2 Purpose

This document describes the interface used to communicate between Cisco BroadWorks servers and partner network elements, including Application Servers, softswitches, Network Servers, SIP proxies, and so on. The network elements are assumed to be trusted and protected by security protocols, firewalls, and/or other security measures. The protocol used by Cisco BroadWorks to communicate between network elements is the Session Initiation Protocol (SIP). This document describes Cisco BroadWorks use of SIP to communicate between network elements. It details the SIP functions implemented by Cisco BroadWorks and enumerates the extensions supported and/or required by Cisco BroadWorks. Additionally, it provides clarification to the SIP specification where required.

3 Specifications

Cisco BroadWorks uses the following specifications for interface-to-network partner solutions:

- RFC 1889: RTP: A Transport Protocol for Real-Time Applications, January 1996
- RFC 1890: RTP Profile for Audio and Video Conferences with Minimal Control, January 1996
- RFC 2327: SDP: Session Description Protocol, April, 1998
- RFC 2806: URLs for Telephone Calls, April, 2000
- RFC 2833: RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals, May 2000
- RFC 2976: The SIP INFO Method, October 2000
- RFC 3261: SIP: Session Initiation Protocol, June 2002
- RFC 3262: Reliability of Provisional Responses in SIP, June 2002
- RFC 3263: Session Initiation Protocol (SIP): Locating SIP Servers, June 2002
- RFC 3264: An Offer/Answer Model with the Session Description Protocol, June 2002
- RFC 3265: Session Initiation Protocol (SIP)-Specific Event Notification, June 2002
- RFC 3266: Support for IPv6 in Session Description Protocol (SDP), June 2002
- RFC 3311: The Session Initiation Protocol (SIP) UPDATE Method, September 2002
- RFC 3312: Integration of Resource Management and Session Initiation Protocol (SIP), October 2002
- RFC 3323: A Privacy Mechanism for the Session Initiation Protocol (SIP), November 2002
- RFC 3325: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, November 2002
- RFC 3326: The Reason Header Field for the Session Initiation Protocol (SIP)", RFC 3326, December 2002
- RFC 3372: Session Initiation Protocol for Telephones (SIP-T): Context and Architectures, September 2002
- RFC 3398: Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping, December 2002
- RFC 3428: Session Initiation Protocol (SIP) Extension for Instant Messaging, December 2002
- RFC 3455: Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP), January 2003
- RFC 3550: RTP: A Transport Protocol for Real-Time Applications, July 2003
- RFC 3551: RTP Profile for Audio and Video Conferences with Minimal Control, July 2003
- RFC 3725: Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP), April, 2004

- RFC 3840: Indicating User Agent Capabilities in the Session Initiation Protocol (SIP), August, 2004
- RFC 3842: A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP), August 2004
- RFC 3959: The Early Session Disposition Type for the Session Initiation Protocol (SIP), December 2004
- RFC 3960: Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP), December 2004
- RFC 3966: The tel URI for Telephone Numbers, December 2004
- RFC 3984: RTP Payload Format for H.264 Video, February 2005
- RFC 4028: Session Timers in the Session Initiation Protocol (SIP), April 2005
- RFC 4244: An Extension to the Session Initiation Protocol (SIP) for Request History Information, November 2005
- RFC 4412: Communications Resource Priority for the Session Initiation Protocol (SIP), February 2006
- RFC 4458: Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR), April 2006.
- RFC 4566: SDP: Session Description Protocol, July 2006
- RFC 4574: The Session Description Protocol (SDP) Label Attribute, August 2006
- RFC 4694: Number Portability Parameters for the "tel" URI, October 2006
- RFC 4904: Representing Trunk Groups in tel/sip Uniform Resource Identifiers (URIs), June 2007
- RFC 5009: Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media
- RFC 5168: XML Schema for Media Control, March 2008
- RFC 5503: Private Session Initiation Protocol (SIP) Proxy-to-Proxy Extensions for Supporting the PacketCable Distributed Call Signaling Architecture, March 2009
- RFC 5806: Diversion Indication in SIP, March 2010[1]
- RFC 5923: Connection Reuse in the Session Initiation Protocol (SIP)
- RFC 5952: A Recommendation for IPv6 Address Text Representation, August 2010
- RFC 5954: Essential Correction for IPv6 ABNF and URI Comparison in RFC 3261, August 2010
- RFC 6044: Mapping and Interworking of Diversion Information between Diversion and History-Info Headers in the Session Initiation Protocol (SIP), October 2010.
- RFC 6086: Session Initiation Protocol (SIP) INFO Method and Package Framework, January 2011
- RFC 6665: SIP-Specific Event Notification, July 2012.
- RFC 6947: The Session Description Protocol (SDP) Alternate Connectivity (ALTC) Attribute, May 2013

[1] Replaces *draft-levy-sip-diversion-08.txt*: Diversion Indication in SIP, August 25, 2004.

- RFC 7315: Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3GPP, July 2014
- draft-mahy-iptel-cpc-00: The Calling Party's Category tel URI Parameter, June 22, 2003
- draft-ietf-sip-privacy-03.txt: SIP Extensions for Caller Identity and Privacy, November 21, 2001
- draft-ietf-sip-privacy-00.txt: SIP Extensions for Caller Identity and Privacy, November 2000
- draft-ietf-avt-rfc2429-bis-04.txt: RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+), December 30, 2004
- draft-poetzl-sipping-call-completion-02: Extensions to the Session Initiation Protocol (SIP) for the support of the Call Completion Services for the European Telecommunications Standards Institute, February 2007
- draft-york-sipping-p-charge-info-06: P-Charge-Info – A Private Header (P-Header) Extension to the Session Initiation Protocol (SIP), February 2009
- draft-portman-siprec-protocol-03: The SIP-based Media Recording Protocol (SIPREC), March 2011
- draft-ram-siprec-metadata-format-01: Session Initiation Protocol (SIP) Recording Metadata Format, March 2011
- ITU-T T.38 Procedures for Real-Time Group 3 Facsimile Communication over IP Networks, September 2005
- NENA i2 Specification: National Emergency Number Association (NENA) Interim VoIP Architecture for Enhanced 9-1-1 Services (i2) Issues 1 Draft, August 5, 2005
- GSMA IR.94 IMS Profile for Conversational Video Service, Version 10.0. October 2015

3.1 Session Initiation Protocol (RFC 3261)

Cisco BroadWorks supports all standard Session Initiation Protocol (SIP) functionality. Following are highlights and clarifications of this support:

- Cisco BroadWorks is implemented as a back-to-back user agent, redirect server, and a registrar.
- Cisco BroadWorks interworks with user agents, redirect servers, and proxy servers.
- Cisco BroadWorks does not support the register mechanism with network devices, but does support the register mechanism with access devices (that is, phone, soft clients, and so on).
- Cisco BroadWorks does not support authentication with network devices, but does support authentication with access devices.
- Cisco BroadWorks supports both UDP and TCP transports.
- Cisco BroadWorks is considered a trusted node in the network and is assumed to have secure connections to other network devices. The Cisco BroadWorks access interface is considered untrusted.
- Cisco BroadWorks supports receiving SIP URIs and TEL URIs.
- Cisco BroadWorks does not tandem/proxy calls.
- All calls received by Cisco BroadWorks must be destined for a Cisco BroadWorks user.
- Cisco BroadWorks rejects calls not destined for a Cisco BroadWorks user by returning a 404 (User Not Found) response.

3.1.1 Support of OPTIONS Method

Cisco BroadWorks uses the OPTIONS method to determine connectivity of devices on the network interface. It is used as an application layer ping to detect SIP responsiveness of the device.

Network devices must support receiving the OPTIONS method. The network device must provide a SIP response to the OPTIONS method. However, the network device does not have to respond with a *200 OK* to the options method. The device may respond with any SIP response code, although a *200 OK* is preferred. Note that Cisco BroadWorks does not include SDP capabilities in the OPTIONS request sent to devices.

Cisco BroadWorks supports receiving the OPTIONS request and responds with a *200 OK*. Note that Cisco BroadWorks does not include SDP capabilities in the OPTIONS response.

3.1.2 Support of SIP over TCP (RFC 3263/ RFC 5923)

Cisco BroadWorks supports TCP as a transport for SIP signaling. *RFC 3261* specifies the behavior for SIP when using a reliable transport protocol. *RFC 3263* provides additional details on determining the transport to use and how to handle failures. The following sections provide clarification on Cisco BroadWorks TCP implementation.

3.1.2.1 Differences between UDP and TCP Transports for SIP

Following are some of the specific differences between TCP and UDP transport for SIP:

- Response handling: When using TCP, responses should be sent using the existing connection to the source of the original request that created the connection. Upon transport failure to send response via TCP, an attempt is made to re-open the connection to the IP address in the *received* parameter, if present, using the port in the "sent-by" value or the default port for that transport, if no port is specified. No forking is performed to attempt to send the response to additional addresses should this fail.
- Transport determination: Following is a summary of the rules from *RFC 3263* section 4.1 in determining the appropriate transport to use for the SIP transactions. Rules are executed in order. Note that Cisco BroadWorks is not currently supporting TLS transport, so those rules have been omitted from the summary.
  - If the URI specifies a transport protocol in the *transport* parameter, that transport protocol should be used.
  - If target is an IP address, UDP should be used.
  - If target is not an IP address, but a port is provided, UDP should be used.
  - If target is not an IP address and no port is provided, a Naming Authority Pointer (NAPTR) lookup is performed for the target. A NAPTR service field of "SIP+D2T" indicates TCP; "SIP+D2U" indicates UDP. The preference and order of the NAPTR record are used to determine the transport. The NAPTR record regular expression field is ignored. Only records with "s" and "a" flags are used. If two NAPTR records have the same preference and order, TCP is used. Note that advancing is not performed between multiple NAPTR records.
  - If no NAPTR records are found, a Service Locator (SRV) query is done for TCP.
  - If no TCP SRV records are found, a UDP SRV lookup is performed.
  - If no UDP SRV records are found, an A record lookup is performed.
- Content-length: TCP requires a *Content-length* header to be included with a value of "zero" when no message body is provided.

When Cisco BroadWorks uses TCP transport, it includes the *transport=tcp* parameter in the Contact entry. Inclusion of the parameter provides better interoperability with devices unable to perform NAPTR or SRV queries to recognize that Cisco BroadWorks prefers the continued use of TCP. However, including this parameter introduces potential deployment limitations when a proxy using TCP does not remain within the dialog (that is, does not Record-Route) and the device on the other side of the proxy does not support TCP. The addition of TCP within the Contact entry also causes the preferred transport to be reflected by NAPTR, if used, and the transport desired by the other device to be disregarded.

Cisco BroadWorks includes the *transport* parameter within the *Request-URI* and URI in the *Route* header as per configuration of the device location within Cisco BroadWorks. When providing a contact within a request and within 18x and 2xx responses, Cisco BroadWorks explicitly includes transport=tcp when sending the message over TCP.

Cisco BroadWorks can also be configured to add transport=udp or transport=tcp to its Contact entries.

uri-parameters = *(";" uri-parameter)

uri-parameter = **transport-param** | user-param | method-param | ttl-param | maddr-param | lr-param | other-param

transport-param = "**transport=**" ("**udp**" | "**tcp**" | "sctp" | "tls" | other-transport)

other-transport = token

The following are examples of the parameters.

INVITE sip:+12405550000@devices.broadworks.net;user=phone;**transport=tcp** SIP/2.0
Route: <sip:devices.broadworks.net;lr;**transport=tcp**>
Contact: <sip:ascluster.broadworks.net;**transport=tcp**>

3.1.2.2 TCP Connection Management

Within *RFC 3261*, devices using a connection-oriented protocol such as TCP typically originate a connection from an ephemeral port. *RFC 3261* provides mechanisms to ensure that responses to a request and new requests sent in the original direction reuse the existing TCP connection. However, as pointed out in *RFC 5923*, new requests sent in the opposite direction more than likely do not reuse the existing connection, causing a connection to be set up in each direction for the call.

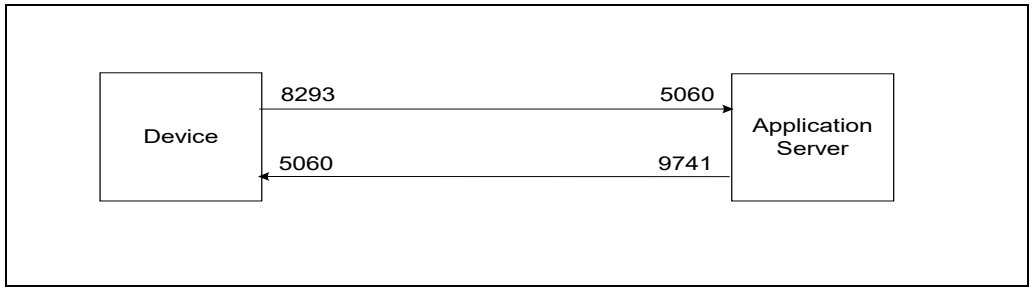

Figure 1 TCP Connection Management

Cisco BroadWorks does not fully support *RFC 5923*. However, Cisco recommends devices that support TCP to reuse the existing connections within a dialog.

It is recommended that all devices support the following requirements when utilizing the TCP transport with Cisco BroadWorks.

Requirements

(R-1) The device must not actively close any sockets unless:

(R-1) a. An error is encountered on the socket.

(R-1) b. A connection has become stale (determined by configuration on the device).

(R-1) c. The maximum number of SIP/TCP sockets supported on the device has been reached, and resources must be reclaimed.

(R-1) d. The maximum number of SIP/TCP sockets per peer supported on the device has been reached, and resources must be reclaimed.

(R-1) e. The device is undergoing maintenance requiring the sockets to be terminated.

(R-2) The device should reuse sockets whenever possible:

(R-2) a. All SIP Responses must be sent on the socket of the corresponding Request.

(R-2) b. After resolving the destination IP address and port, but before sending any SIP Request, the device should determine whether an existing connection has already been established to the destination IP address and port. If an existing connection is found, the device should reuse the connection. If an existing connection is not found, the device may initiate a new connection.

3.1.3 SIP Timers

Cisco BroadWorks implements the various SIP timers defined in *RFC 3261*. The following table describes Cisco BroadWorks-specific behavior.

Timer	Comment
T1	T1 has a default value of 500 ms but can be configured to one of 500 ms, 1 second, 2 seconds, 5 seconds, 7 seconds, or 9 seconds.
T2	T2 has a default value of 4 seconds but can be configured to one of 4 seconds, 6 seconds, 8 seconds, or 10 seconds.
T4	Cisco BroadWorks does not use this timer parameter (see Timer I and Timer K in this table).
Timer A	Under normal conditions, Cisco BroadWorks uses Timer A as defined in *RFC 3261*. Cisco BroadWorks initializes Timer A to T1 and doubles at every retransmission. Under heavy load conditions, Cisco BroadWorks' behavior changes in the following ways: ▪ When Cisco BroadWorks is in Yellow CallP Overload condition, it initializes Timer A to 2*T1. ▪ When Cisco BroadWorks is in Red CallP Overload condition, it initializes Timer A to 4*T1.

Timer B	For TCP, Cisco BroadWorks uses Timer B according to *RFC 3261*. Under normal conditions, Cisco BroadWorks initializes Timer B to 64*T1.
	For UDP, Cisco BroadWorks uses a retry count according to *RFC 2543* instead of using Timer B. The number of tries for INVITE is 7; however, if the doubling of time between tries exceeds 32 seconds, it becomes the final retry to the location.
	Cisco BroadWorks can route advance a request to multiple IP address and port locations according to *RFC 3263*. In this situation, Cisco BroadWorks may use a shorter Timer B value for all locations except for the last, based on the *suspiciousAddressThreshold* system parameter (if configured).
	▪ When the transport is UDP, Cisco BroadWorks interprets the *suspiciousAddressThreshold* as the number of delivery attempts before advancing to the next location.
	▪ When the transport is TCP, Cisco BroadWorks uses the *suspiciousAddressThreshold* to compute the value of Timer B as follows:
	  ▪ If *suspiciousAddressThreshold* is 1, the value is "T1".
	  ▪ If *suspiciousAddressThreshold* is greater than 1, then Cisco BroadWorks determines the Timer B value according to a recursive formula. Specifically, if *suspiciousAddressThreshold* is set to n, the value for Timer B is the value for (n – 1) plus the minimum of T2 and 2^(n-1)*T1. (This is similar to the non-INVITE retry rule defined in *RFC 3261*).
	▪ When Cisco BroadWorks is in Yellow CallP Overload condition, it adds an extra 2 seconds to Timer B.
	▪ When Cisco BroadWorks is in Red CallP Overload condition, it adds an extra 4 seconds to Timer B.
	Regardless of the above, the maximum value is 32 seconds.
	Note that the same logic also applies to Timer F.

Timer C	Cisco BroadWorks does not use Timer C because it does not act as a stateful proxy. (Cisco BroadWorks is a B2BUA.)

Timer	Comment
Timer D	The wait time for late retransmission is handled differently in Cisco BroadWorks. Cisco BroadWorks remembers received messages and keeps them in memory until it runs the stale message audit. This audit is run at most once every 10*T2. When it executes, messages older than 10*T2 are removed. This is equivalent to Timer D firing after an interval of at least 10*T2, but possibly 20*T2 or more, for both UDP and TCP. Furthermore, this time can be shortened if the SIP dialog is released. ▪ If the *useSessionCompletionTimer* system parameter is true, the dialog post-released retention period is configured using the *sessionCompletionTimer* system parameter value (5 to 100 seconds). ▪ If the *useSessionCompletionTimer* system parameter is false, the dialog post-released retention period is set to "10*T2" under normal conditions. When Cisco BroadWorks is in Yellow or Red CallP Overload condition, dialog post-released retention is forced to 5 seconds. Note that the same logic also applies to Timers I, J, and K.
Timer E	Cisco BroadWorks uses Timer E as defined in *RFC 3261*. Cisco BroadWorks initializes Timer E to T1 and doubles at every retransmission until it reaches a maximum value of T2. Cisco BroadWorks does not start using interval T2 upon receiving a 1xx response; the purpose of this *RFC 3261* non-compliance is to accommodate devices that are not compliant with *RFC 4320*. Under heavy load conditions, Cisco BroadWorks' behavior changes as follows: ▪ When Cisco BroadWorks is in Yellow CallP Overload condition, it initializes Timer E to 2*T1. ▪ When Cisco BroadWorks is in Red CallP Overload condition, it initializes Timer E to 4*T1.
Timer F	Cisco BroadWorks uses Timer F as defined in *RFC 3261* for TCP. It is initialized to 64*T1. For UDP, Cisco BroadWorks uses a retry count according to *RFC 2543* instead of using Timer F. According to *RFC 2543*, the number of tries for a non-INVITE request is 11. Cisco BroadWorks can route advance a request to multiple IP address and port locations according to *RFC 3263*. In this situation, Cisco BroadWorks may use a shorter Timer F value. For a description on how this shorter time is computed, see the comment for Timer B.
Timer G	Cisco BroadWorks uses Timer G as defined in *RFC 3261*. Under normal conditions, Cisco BroadWorks initializes Timer G to T1 and doubles it at every retransmission. However, Cisco BroadWorks does not use T2 as the maximum interval between retransmissions. Cisco BroadWorks also uses this timer for retransmitting INVITE 2xx responses and reliable 1xx responses. Under heavy load conditions, Cisco BroadWorks' behavior changes as follows: ▪ When Cisco BroadWorks is in Yellow CallP Overload condition, it initializes Timer G to 2*T1. ▪ When Cisco BroadWorks is in Red CallP Overload condition, it initializes Timer G to 4*T1.
Timer H	Cisco BroadWorks uses Timer H as defined in *RFC 3261* for TCP (unless retrying a response as described for Timer G). Cisco BroadWorks initializes Timer H to 64*T1. For UDP and TCP, when retrying according to Timer G, Cisco BroadWorks uses a retry count according to *RFC 2543* instead of using Timer H. The number of tries for INVITE responses is 7; however, if the doubling of time between tries exceeds 32 seconds, it becomes the final retry to the location. When deployed within a Session Data Replication redundancy model, Cisco BroadWorks can route advance an INVITE response to multiple IP address and port locations according to *RFC 3263*. In this situation, Cisco BroadWorks may advance after fewer attempts for all locations except for the last, based on the *suspiciousAddressThreshold* configuration (if configured).
Timer I	The wait time for late retransmission is handled differently in Cisco BroadWorks. See the comment for Timer D.

Timer	Comment
Timer J	The wait time for late retransmission is handled differently in Cisco BroadWorks. See the comment for Timer D.
Timer K	The wait time for late retransmission is handled differently in Cisco BroadWorks. See the comment for Timer D.

3.1.1 Quick re-INVITE Delay

Service execution sometimes requires Cisco BroadWorks to send an ACK request followed immediately by a re-INVITE request. In certain deployments, intermediary servers may reorder these messages. For example, when the ACK request has an SDP and the INVITE request does not, a Proxy Call Session Control Function (P-CSCF) can perform extra processing steps on the ACK request causing a delay, while forwarding the INVITE immediately. Such processing can result in the requests being out of order at the endpoint.

Two system parameters control this behavior to enable a defined delay to prevent INVITEs to be sent too closely following an ACK. Reordering may still occur; however, this timer aims at reducing its frequency.

When this system-level SIP configuration (*enableDelayQuickReInvite*) is enabled, the *delayQuickReInviteMilliseconds* system-level SIP parameter is read to determine the amount of time (delay) an Application Server should wait until the INVITE message is sent following an ACK. By default, the value is "1000" milliseconds (ms), but can be configured to between 100 ms and 10 seconds.

The Application Server adds this delay for certain re-INVITEs sent quickly after an ACK. This includes scenarios in which a service in the Application Server originates an ACK followed by a re-INVITE. It also includes scenarios where the Application Server proxies messages from one end to the other.

Note that it is not suggested to enable this configuration unless interoperability issues take place and cannot be avoided since it introduces inescapable delays every time an INVITE is sent quickly after an ACK.

3.1.2 Call-ID Suffix

By default, Cisco BroadWorks builds the value of the *Call-ID* header using the following format:

BW + Time based String + Random Number + @ + AS IP Address or FQDN

Examples

```
Call-ID: BW164437260180913-2005069729@192.168.8.193
Call-ID: BW164522011180913-1788629683@ascluster.example.net
```

This format may expose the server network address externally and can be a security issue. To avoid exposing the server address, Cisco BroadWorks allows an administrator to change the suffix of the *Call-ID* header value.

If an administrator sets a value for the start-up parameter *bw.sip.callidSuffix*, then Cisco BroadWorks uses that value as the suffix for the *Call-ID* header value, replacing the domain part. The format for the *Call-ID* header value is then the following:

BW + Time based String + Random Number + @ + Custom Call-ID Suffix

Examples

```
Call-ID: BW164437260180913-2005069729@EXAMPLE
```

```
Call-ID: BW164522011180913-1788629683@as.cluster
```

Because this suffix parameter is a start-up parameter, each individual server in an Application Server cluster has its own value.

3.1.2.1 RFC 2543 and RFC 3261 Compatibility Recommendation

RFC 2543 and *RFC 3261* define the *Call-ID* header syntax differently. To be compatible and interoperable with both RFCs, Cisco BroadWorks enforces the provisioning of the custom *Call-ID* suffix with alphanumeric characters, dashes, and dots only (with a minimum length of 1 valid character).

Examples of Call-ID suffixes compatible with both RFC 2543 and RFC 3261

```
Call-ID: BW164437260180913-2005069729@EXAMPLE1-CLUSTER2-SUFFIX3
Call-ID: BW164522011180913-1788629683@singletoken
```

3.1.2.2 P-Charging-Vector Impact

Cisco BroadWorks builds the *P-Charging-Vector* header using the Application Server IP address or FQDN the same way as it does the *Call-ID* header. Setting the *Call-ID* suffix impacts the value of the *P-Charging-Vector* (PCV's) IMS Charging Identity (ICID) component. When setting the new *Call-ID* suffix, the global uniqueness of the ICID must be considered.

3.1.3 Inter-Cluster Spiraling

The Application Server's spiral detection mechanism checks the *Via* branch parameter to determine when a SIP request spirals toward itself or its cluster mate. By default, Cisco BroadWorks builds this parameter using either the *bw.sip.accessinterfaceviahost* start-up parameter for stand-alone deployments or the IP address of the primary peer in redundancy deployments. Cisco BroadWorks encodes this value then appends it to the magic cookie to form the value of the *branch* parameter. The following example shows the default format:

branch=z9hG4bKBroadWorks.1jmomaf

z9hG4bK is the magic cookie (RFC 3261).

BroadWorks. is the prefix.

1jmomaf is the encoded token identifying a unique Application Server (stand-alone) or Application Server cluster (redundant).

When a user hosted on an Application Server cluster calls another user hosted on another Application Server cluster (on the same network) who performs a Call Forward to a user hosted on the same cluster as the originator, the call may not complete as expected. The default spiral-detection mechanism may fail to identify the INVITE request sent from the other cluster as a termination. This failure results in a second originating call to be created for the calling party with the side effect of running originating services again and negatively impacting billing. To avoid this problem, an administrator can configure Cisco BroadWorks to use a different token in the *Via* header's *branch* parameter. When this alternate token is configured, the Application Servers in the same cluster have the same token, and Cisco BroadWorks is able to detect the spiral.

If an administrator sets the value of the SIP parameter *viaBranchToken*, then Cisco BroadWorks uses that value instead of the usual encoded token. If *viaBranchToken* has no value, then builds the *branch* parameter value using the default format described previously.

This configuration is not to be used in IMS mode. This functionality applies only in stand-alone Application Server deployments where the presentation identity is the line/port.

3.1.3.1 Private Branch Exchange Consideration

When an Application Server functions as a Private Branch Exchange (PBX), the value of the custom token, if configured, should be different from the value of the Application Server that functions as the hosting server for the served trunk users. In such deployments, the system operator is responsible to ensure that both tokens are different. The Application Server does not do this automatically. Failure to configure different values results in originating calls being incorrectly identified as terminations and thus failures.

3.1.3.2 Interoperability

To ensure interoperability with other network elements, the value of this new parameter is limited to use alphanumeric characters up to a length of seven characters.

3.2 SIP Subscriber Identification/Addressing

Cisco BroadWorks supports the following:

- URLs for telephone calls (*RFC 2806*)
- The tel URI for telephone numbers (*RFC 3966*)
- *NP* parameters for the tel URI (*RFC 4694*)
- The calling party's category tel URI parameter (*draft-mahy-iptel-cpc-0*0)
- A Privacy Mechanism for the Session Initiation Protocol (SIP) (*RFC 3323*)
- Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks (*RFC 3325*)
- SIP Extensions for Caller Identity and Privacy (*draft-ietf-sip-privacy-03, draft-ietf-sip-privacy-00)*
- Diversion Indication in SIP (*RFC 5806*)

Cisco BroadWorks supports all standard SIP functionality for addressing, as specified in *RFC 3261*. Cisco BroadWorks also supports a number of specifications related to subscriber identification and addressing in SIP. Following are highlights and clarifications of this support:

- Addressing information:
  - Cisco BroadWorks can optionally encrypt the *FROM* header information to protect the privacy of the calling party.
  - Cisco BroadWorks always includes the *P-Asserted-Identity* header or *Remote-Party-ID* header for all calls sent by Cisco BroadWorks on the network interface, based on the *privacyVersion* setting on the Application Server.
  - Cisco BroadWorks supports called party identity including name and number. Cisco BroadWorks supports both Tel URIs and SIP URIs. Cisco BroadWorks also supports the telephone-subscriber contained in the SIP URI, including all proper escaping.

- For INVITES sent from a Cisco BroadWorks server.

Header	Format	Parameter	Value
Request-URI	SIP URI	User	Dialed number or E.164 number.
		Host	Network device domain name or IP address.
From		display-name	User or group name.
	SIP URI	User	User or group phone number. Number is in E.164 format if Cisco BroadWorks is configured to send in E.164 format. Otherwise, the number is a national number.
		Host	Application Server cluster domain name, subscriber domain name, or IP address.
To	SIP URI	User	Dialed number or E.164 number.
		Host	Network device domain name or IP address.
Contact	SIP URI	User	Not used (empty).
		Host	Application Server cluster domain name or IP address.
Remote Party ID		display-name	User or group name.
	SIP URI	User	User or group phone number. Number is in E.164 format if Cisco BroadWorks is configured to send in E.164 format. Otherwise, the number is a national number.
		Host	Application Server cluster domain name, subscriber domain name, or IP address.
P-Asserted-Identity		display-name	User or group name.
	SIP URI	User	User or group phone number. Number is in E.164 format if Cisco BroadWorks is configured to send in E.164 format. Otherwise, the number is a national number.
		Host	Application Server cluster domain name, subscriber domain name, or IP address.

- For INVITEs sent from a network device:

Header	Format	Parameter	Value
Request-URI	SIP URI	User	Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
		Host	Network device domain name, IP address, Application Server cluster domain name, Application Server alias, or subscriber domain name.

Header	Format	Parameter	Value
	Tel URI		Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
From		display-name	Calling party name, if available.
	SIP URI	User	Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
		Host	Network device domain name or IP address.
	Tel URI		Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
To	SIP URI	User	Should be dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context. However, Cisco BroadWorks does not perform any translations on the *To* header SIP URI. Therefore, the *To* header SIP URI may contain anything.
		Host	Network device domain name, IP address, Application Server cluster domain name, Application Server alias, or subscriber domain name.
	Tel URI		Should be dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context. However, Cisco BroadWorks does not perform any translations on the *To* header Tel URI. Therefore, the *To* header SIP URI may contain anything.
Contact	SIP URI	User	Anything as allowed per the specification.
		Host	Network device domain name or IP address.
Remote Party ID		display-name	Calling party name, if available.
	SIP URI	User	Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
		Host	Network device domain name, IP address, Application Server cluster domain name, Application Server alias, or subscriber domain name.
	Tel URI		Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
P-Asserted-		display-name	Calling party name, if available.

Header	Format	Parameter	Value
Identity	SIP URI	User	Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
		Host	Network device domain name, IP address, Application Server cluster domain name, Application Server alias, or subscriber domain name.
	Tel URI		Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
P-Preferred-Identity		display-name	Calling party name, if available.
	SIP URI	User	Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.
		Host	Network device domain name, IP address, Application Server cluster domain name, Application Server alias, or subscriber domain name.
	Tel URI		Dialed number/national number, E.164 number, or dialed number with global or local network prefix phone-context.

Cisco BroadWorks application optionally adds a *classmark* parameter to the telephone subscriber part of the *From* header for origination or redirection to the PSTN.

```
classmark = ";classmark=" 1*alphanum
```

Example:

```
From:"test"<sip:+13015558888;classmark=123@domain.net;user=phone>;tag=51-
```

3.3 URLs for Telephone Calls (RFC 2806)/The tel URI for Telephone Numbers (RFC 3966)

Cisco BroadWorks fully supports the tel URI scheme, as specified in *RFC 2806*. A network device may use either a tel URI or a SIP URI in requests sent to Cisco BroadWorks.

Cisco BroadWorks only supports the revised *RFC 3966* in subsequent releases.

3.4 Privacy Mechanism for SIP/Private Extensions to SIP for Asserted Identity within Trusted Networks (RFC 3323/RFC 3325)

Cisco BroadWorks supports the standards track *RFC 3323, A Privacy Mechanism for the Session Initiation Protocol (SIP)*, and *RFC 3325, Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks*, to protect the identity of Cisco BroadWorks users when interworking with network devices. Cisco BroadWorks also supports alternative privacy mechanisms defined by older IETF draft documents related to privacy, such as *draft-ietf-sip-privacy-00* and *draft-ietf-sip-privacy-03*. Cisco BroadWorks uses provisioning to determine which privacy implementation to use when sending INVITEs.

Cisco BroadWorks does not explicitly define a "Spec(T)" as described in *RFC 3325*. Cisco BroadWorks considers all devices on the network interface as trusted entities. The Network Interface Access Control list and Network Traffic Security feature are used to maintain the integrity of the network interface since all devices on the network interface are trusted. The access interface is treated as untrusted and all devices on the access interface are not considered part of the trusted network.

RFC 3325 adds two new SIP headers: *P-Asserted-Identity* and *P-Preferred-Identity*. *RFC 3325* specifies that a party in a call dialog can be assigned one or more explicit identities within a trust domain indicated by one or more *P-Asserted-Identity* headers. These headers supersede any other headers when it comes to matching a user.

Cisco BroadWorks supports only one identity for a given user in a SIP message. This identity includes a user name and can also include a display name. According to *RFC 3325*, multiple *P-Asserted-Identity* headers are allowed within a SIP message. However, Cisco BroadWorks only acts upon the first one encountered in any SIP message that matches a Cisco BroadWorks originator. Note that if one or more *P-Preferred-Identity* headers are present, Cisco BroadWorks uses the first *P-Preferred-Identity* header encountered, to attempt to identify the Cisco BroadWorks subscriber. Cisco BroadWorks supports only one identity for a given user in a SIP message. This identity includes a username and can also include a display name.

When an invitation is sent to a trusted (network) device, Cisco BroadWorks always includes a *P-Asserted-Identity* header in the initial **INVITE** message. If "id" privacy is requested in the *Privacy* header, then no *P-Asserted-Identity* is included in any invitation destined for a non-trusted (access) device. Otherwise, a *P-Asserted-Identity* header is included just as in the case where the message is sent to a trusted device. Cisco BroadWorks never populates an outgoing message with a *P-Preferred-Identity* header.

Cisco BroadWorks acts as a Privacy service as described in *RFC 3323*. *RFC 3323* adds a new SIP header called *Privacy*. It can be assigned one or more of the following values: "none", "header", "session", "user", and "critical". "header", "session", and "user" indicate specific areas within SIP messages where privacy should be enforced. Similarly, *RFC 3325* adds "id" as a new area of privacy and *RFC 4244* adds "history". The values "none" and "critical" are defined in *RFC 3323*. If neither "none" nor "critical" is present, then the device is instructed to give its best effort to achieve privacy.

Cisco BroadWorks supports the values of the *Privacy* header as follows:

- "None"
  - "None" indicates that the Privacy service must not add privacy. In terms of Cisco BroadWorks, this means that Calling Line ID (CLID) blocking should be disabled and the identity of the originator should be made available. Cisco BroadWorks honors this value on the network interface. However, on the access interface, Cisco BroadWorks *do not* honor this value and use the setting of the Cisco BroadWorks subscriber CLID Delivery Blocking service to determine whether the identity should be presented.
- "Critical"
  - "Critical" indicates that the message should be rejected if the Privacy service cannot or will not enforce the specified privacy. If one of the values of the *Privacy* header is "critical" and Cisco BroadWorks chooses not to honor that privacy request, Cisco BroadWorks rejects the invitation with a SIP *500 Server Error* response as specified in *RFC 3323*. Note that this choice of return value is specified in *RFC 3323* with strength of "MUST".
- "User"
  - "User" indicates that the Privacy service should enforce user-level privacy for the subscriber, by removing any user identification from the SIP message. Cisco BroadWorks treats the request privacy as requesting "Anonymous Presentation". There is no mechanism in *RFC 3323* to specify only one of "Anonymous Name" or "Anonymous URI" as there was in previous drafts.

    Cisco BroadWorks handles "Anonymous Presentation" by replacing the value of the *NameAddr* in the *From* header with the following:

    "Anonymous" <sip:anonymous@anonymous.invalid>

    Note that this is different from the NameAddr format used in previous releases to hide identity if the proprietary privacy support was configured:

    "Anonymous" <sip:Private@localhost>

    Cisco BroadWorks sends the new *NameAddr* value for all scenarios, regardless of whether or not this new version of privacy is currently configured for use in Cisco BroadWorks.
- "Header"
  - "Header" indicates that the Privacy service should enforce privacy for all of the headers in the SIP message that may identify information about the subscriber. This includes the *Via* and *Contact* headers. Cisco BroadWorks provides this capability by default as a back-to-back user agent.
  - Cisco BroadWorks sends the "header" value if the presentation indicator for a call indicates "private" and the SIP system parameter *includeHeaderLevelPrivacyParameter* is set to "true".
  - The SIP system parameter *supportHeaderLevelPrivacy* controls whether Cisco BroadWorks should apply anonymous presentation for calling user's identity in response to receiving the "header" value. If *supportHeaderLevelPrivacy* is set to "true", then Cisco BroadWorks applies anonymous presentation in the same way it does for "user" or "id" privacy. If *supportHeaderLevelPrivacy* is set to "false", then Cisco BroadWorks does not apply anonymous presentation (unless the "user" or "id" value is also present in the *Privacy* header).

- "Session"
  - "Session" indicates that the information held in the Session Description (SDP) should be hidden outside of the trust domain. To accomplish this, a Privacy service would have to terminate the session on one end and originate one on the other end.
  - If Cisco BroadWorks receives a request that includes a *Privacy* header that indicates both "session" and "critical", Cisco BroadWorks rejects the message with a *500 Server Error* response.
- "Id"
  - The Asserted Identity *RFC 3325* adds the value "Id" to the *Privacy* header. In such a case, the *P-Asserted-Identity* header is removed upon leaving the trust domain. In other words, if a message from the PSTN contains a *P-Asserted-Identity* header and a *Privacy* header that contains "id", then the *P-Asserted-Identity* header is not populated when the invitation is sent to an access device.
- "History"
  - The History-Info *RFC 4244* adds the value "history" to the *Privacy* header. This requests that privacy be applied to the *History-Info* header or to specific entries when sending the header outside the trusted domain. For more information on History privacy, see section *3.10 Cisco BroadWorks Support for Request History*.

For RFC 3323/3325 privacy:

- For INVITEs sent from a Cisco BroadWorks server where the calling party identity is restricted:
  - Privacy Header = id, critical
  - P-Asserted-Identity = set the same as the *From* header when the *From* header is not encrypted
- For INVITEs sent from a network device where the calling party identity is restricted:
  - Privacy Header = id, critical
  - P-Asserted-Identity and/or P-Preferred-Identity = set the same as the *From* header when the *From* header is not encrypted

3.5 SIP Extensions for Caller Identity and Privacy (draft-ietf-sip-privacy-03, draft-ietf-sip-privacy-00)

Cisco BroadWorks supports the SIP Extensions for Caller Identity and Privacy draft to protect the identity of Cisco BroadWorks users when interworking with network devices. Note that Cisco BroadWorks supports both version 00 and 03 of this draft. Cisco BroadWorks uses provisioning to determine which version of the draft to use when sending INVITEs.

Cisco BroadWorks also supports a privacy mechanism without using the *draft-ietf-sip-privacy-00* or *draft-ietf-sip-privacy-03*. Cisco BroadWorks reserves certain display names in the name-addr of the SIP-URI as keywords to protect the privacy of the calling party.

For draft-ietf-sip-privacy-00:

- For INVITEs sent from a Cisco BroadWorks server where the calling party identity is restricted:
  - Remote-party-ID = set the same as the *From* header when the *From* header is not encrypted

- Anonymity = set based on whether the calling line identity is restricted (off <or> uri, name)
- Cisco BroadWorks does not support *full* or *ipaddr* options of the *Anonymity* header.

- For INVITEs sent from a network device where the calling party identity is restricted:
  - Remote-party-ID = set the same as the *From* header when the *From* header is not encrypted
  - Anonymity = set based on whether the calling line identity is restricted (off <or> uri, name <or> uri <or> name)
  - Cisco BroadWorks does not support *full* or *ipaddr* options of the *Anonymity* header.

For draft-ietf-sip-privacy-03:

- For INVITEs sent from a Cisco BroadWorks server where the calling party identity is restricted:
  - Remote-party-ID = set the same as the *From* header when the *From* header is not encrypted and includes the *rpi-privacy* parameter set to “full” or “off” based on whether the calling line identity is restricted
  - Anonymity = off
  - Cisco BroadWorks does not support the *ipaddr* option of the *Anonymity* header.
- For INVITEs sent from a network device where the calling party identity is restricted:
  - Remote-party-ID = set the same as the *From* header when the *From* header is not encrypted and includes the *rpi-privacy* parameter set to (full <or> uri <or> name <or> off) based on whether the calling line identity is restricted.
  - Anonymity = off
  - Cisco BroadWorks does not support the *ipaddr* option of the *Anonymity* header.

For proprietary privacy support:

- For INVITEs sent from a Cisco BroadWorks server where the calling party identity is restricted:
  - Display name in the *name-addr* of the *From* header is set to “anonymous” when the calling line identity is restricted.
  - The user portion of the SIP-URI in the *From* header is empty when the calling party identity is not available, for example, sip:10.10.124.56, sip:broadsoft.com, and so on.
- For INVITEs sent from a network device where the calling party identity is restricted:
  - Display name in the *name-addr* of the *From* header is set to “anonymous” when the calling line identity is restricted.

    The user portion of the SIP-URI in the *From* header is empty when the calling party identity is not available, for example, sip:10.10.124.56, sip:broadsoft.com, and so on.

3.6 Number Portability Parameters for the tel URI (RFC 4694)

Cisco BroadWorks supports the *cic* parameter for Equal Access calls as defined in the *RFC 4964*. Cisco BroadWorks supports assigning of Preferred Inter-exchange Carriers (PICs) to users, groups, and enterprises (for enterprises, on a per-country code basis). The Cisco BroadWorks Network Server appends the *cic* parameter when the policy is provisioned on the Network Server. The Cisco BroadWorks Application Server passes the *cic* parameter received in the contacts in a 3xx in the Request-URI of the subsequent INVITE sent to the contact specified in the 3xx.

Cisco BroadWorks also includes a lata *telephone-subscriber* parameter for Equal Access calls. The lata parameter is included when the policy is provisioned on the Network Server and the following conditions are met:

- The CIC is known and valid.
- The LATA of the calling party is known and provisioned on the Network Server.

For more information on Cisco BroadWorks Equal Access support, see *Appendix B: Cisco BroadWorks Equal Access Support*.

3.7 Calling Party's Category tel URI Parameter (draft-mahy-iptel-cpc-00)/isup-oli Parameter Support

Cisco BroadWorks supports the *cpc* parameter as specified in *draft-mahy-iptel-cpc-00*. Additionally, Cisco BroadWorks also supports a similar SIP URI parameter, *isup-oli,* and a *CPC* value in a *gtd* message body. These parameters carry originating line information or the calling party category (CPC) of the subscriber for use by softswitches or network gateways. Typically, the parameter value characterizes the originating party, for example, pay phone, prison, hotel, hospital, and so on. In the following description, CPC is used as a generic term to indicate the value populated using the *cpc* parameter, the *isup-oli* parameter, or *CPC* in a *gtd* body.

Cisco BroadWorks only includes the CPC value based on the Cisco BroadWorks SIP network interface configuration. Cisco BroadWorks never includes more than one parameter.

Note that the information populated by Cisco BroadWorks in the *CPC* parameter for North America is populated by a softswitch or network gateway in the *originating line information* parameter in the ISDN User Part Initial Address Message (IAM). The information carried in these parameters may be populated in any appropriate parameter, as determined by the Time Division Multiplexing (TDM) protocol on the softswitch or network gateway.

Cisco BroadWorks populates the *CPC* parameter based on the Cisco BroadWorks subscriber service, calling party category (CPC). The CPC service allows a category to be associated with a subscriber. The category is included in the signaling for all outgoing calls. It is used by a softswitch or switching system for call routing, and it is used by the operator services system to determine the allowed policies for a subscriber.

When the CPC service is assigned to a subscriber, the subscriber is automatically assigned the default CPC, depending on the Cisco BroadWorks provisioning of the network interface.

The CPC value is only included when the call is routed to a PSTN terminating party. When calls are made within a group, the CPC values are not included in the INVITE. Hence, inter-group calls and network calls may have the CPC value included in the SIP INVITE during a call origination, forward, or transfer.

During a call forward or call transfer to another subscriber by the terminating party in a different group, and in such a case as when the terminating party has the CPC service, the CPC value that is substituted in the SIP INVITE to the terminating party and the CDR are the CPC value of the “call forwarder” or the “call transferor”.

Cisco BroadWorks supports the following three CPC formats:

- cpc format
- isup-oli format
- CPC in *gtd* body format

Note that the actual values of *cpc* parameters, *isup-oli* parameters, and CPC in *gtd* bodies are configurable at the CLI.

The following SIP INVITE examples show the *From* header of the SIP INVITE request for both TEL URIs and SIP URIs with both *cpc* and *isup-oli* parameter formats.

```
INVITE sip:bart@biloxi.example.com SIP/2.0
To: "Bart" <sip:bart@biloxi.example.com>
From: <tel:+17005554141;cpc=payphone>;tag=1928301774

INVITE sip:bart@biloxi.example.com SIP/2.0
To: "Bart" <sip:bart@biloxi.example.com>
From: <sip:+17005554141;cpc=payphone@example.net>;tag=1928301774
```

```
INVITE sip:bart@biloxi.example.com SIP/2.0
To: "Bart" <sip:bob@biloxi.example.com>
From: <tel:+17005554141;isup-oli=70>;tag=1928301774

INVITE sip:bart@biloxi.example.com SIP/2.0
To: "Bart" <sip:bob@biloxi.example.com>
From: <sip:+17005554141@example.net;isup-oli=70>;tag=1928301774
```

The following SIP INVITE example shows the GTD body with the CPC value.

```
INVITE sip:bart@biloxi.example.com SIP/2.0
To: "Bart" <sip:bob@biloxi.example.com>
From: <sip:+17005554141@example.net;isup-oli=70>;tag=1928301774
...
Content-Type:application/gtd
Content-Length:24

IAM,
VER,0.18
CPC,15
```

3.8 Incoming/Outgoing OTG Support

Cisco BroadWorks supports the *otg* (originating trunk group) URI parameter in the *From* header. The *otg* parameter, referred to as the sourceId in the Network Server, is another way to identify an enterprise, user group, or site in the system. When a sourceId is chosen to identify an enterprise, user group, or site, it can also be used to identify any other subscriber in the system. A one-to-many mapping therefore, exists between a sourceId and a Network Server subscriber (enterprise, user group, or site).

The sourceId is a string that can contain up to 128 characters, the minimal length being 0 (a 0-length sourceId simply means that the subscriber has no sourceId and does not use the *otg* parameter). The sourceId can only be composed of alphanumerical characters, as specified by *RFC 3261*.

A sourceId can be provisioned at the enterprise, user group, and site levels.

3.8.1 Outgoing OTG

During a call, if the Network Server can identify that the originator has a valid sourceId (that is, the originating entity – enterprise, user group, or site – has a provisioned sourceId in the Network Server), the Network Server populates the *otg* parameter in each *Contact* entry returned in response to the received INVITE request, using the following logic:

Site of Caller	User Group of Caller	Enterprise of Caller	sourceId Used in Outgoing *otg* in 302 Contact Entry
A	Do not care	Do not care	A
Not provisioned	B	Do not care	B
Not provisioned	Not provisioned	C	C
Not provisioned	Not provisioned	Not provisioned	*otg* not added

The *otg* parameter is only added to the *Contact* entry if the 302 message is to be sent back to a hosting network element that supports the *sourceId* signaling option. In addition, the *otg* parameter is only added if the call originator is derived from the *From, P-Asserted-Identity*, or *Remote-Party-ID* header. If the Network Server considers that the call originator is identified by the *Diversion* header, then it does not add the *otg* parameter.

The Network Server does not support direct originating trunk groups in tandem. The *otg* parameter returned in 302 Contact entries is only based on internal processing, regardless of its presence or not in the SIP INVITE request.

3.8.2 Incoming OTG

The incoming OTG capability is only triggered if the Network Server derives the call originator from the *From* or *P-Asserted-Identity* header. If the call originator, as determined by the Network Server, is taken from the *Remote-Party-ID* or *Diversion* header, incoming OTG is not supported.

Once the originating URI is identified, the Network Server uses the *user* and *host* portions of the originating URI to map the caller to a known subscriber in the system. With the incoming *otg* parameter, if no subscriber can be identified from the originating URI, the Network Server finds the subscriber to use, by looking at the value of the *otg* parameter in the call originator header (*From* or *P-Asserted-Identity*) of the SIP INVITE request. If the *otg* parameter is present, the Network Server extracts it to find the corresponding subscriber in the system, using the following logic:

OTG Maps to Site	OTG Maps to User Group	OTG Maps to Enterprise	Public Profile Used	Private Profile Used
Yes	N/A	N/A	Site profile	Site's enterprise private profile
N/A	Yes	N/A	User group profile	User group's enterprise private profile
N/A	N/A	Yes	Enterprise public profile	Enterprise private profile
No	No	No	Same logic as today	Same logic as today

Incoming OTG support is always enabled and cannot be disabled.

3.8.3 Network Server SIP Encoding of otg Parameter

The Network Server encodes the *otg* parameter in *302 Contact* entries as follows.

```
Contact:  <sip:+15146971111@broadsoft.com;user=phone?P-Asserted-
Identity=%3csip:+15146972222%40broadsoft.com%3buser%3dphone%3botg%3dtest%
3e>;q=0.5,<sip:5146971111@192.168.12.5:5060;user=phone?P-Asserted-
Identity=%3csip:+15146972222%40broadsoft.com%3buser%3dphone%3botg%3dtest%
3e>;q=0.25
```

The Network Server returns the sourceId as an *otg* parameter in a *P-Asserted-Identity* header embedded within the URI in each *Contact* entry. The URI included in the *P-Asserted-Identity* is taken from the SIP INVITE request header that the Network Server used to identify the call originator, along with the attached user-part and host-part options, such as *user=*. The special characters in the embedded *P-Asserted-Identity* header are URL encoded (that is, "escaped"), as shown above.

In addition, the Network Server accepts an incoming *otg* parameter in the *From* or *P-Asserted-Identity* header of a SIP INVITE request. Following is an example of the *otg* URI parameter:

```
From: <sip:+15146972222@broadsoft.com;user=phone;otg=test>
```

3.8.4 Application Server Handling of OTG URI Parameter

The Application Server only honors the *P-Asserted-Identity* header parameter included in a 302 response when it comes from a Network Server. In this case, if the Application Server receives a 302 response in which *Contact* entries contain a *P-Asserted-Identity header* parameter, the Application Server replaces the contents of the calling party information with the contents of the *P-Asserted-Identity* including the *otg* URI parameter, if present.

The Application Server then populates the resulting INVITE request with the calling party information obtained from the Network Server. This information is populated into the *From* header, *Remote-Party-ID* header, or *P-Asserted-Identity* header as determined by the Application Server SIP network interface configuration (that is, determined by the *privacyVersion* system parameter).

In addition, the Application Server may populate the *otg* or *tgrp* parameters depending on the trunk group configuration independently from the Network Server response. Two trunk group configurable parameters control this behavior:

- *includeOtgIdentityForNetworkCalls* – When set to "true", trunk group originated calls terminating to the network side include the *otg* parameter in the *From* header using the provisioned data from the *OTG/DTG* field. A pilot user is not required for this to occur.
- *includeTrunkGroupIdentityForNetworkCalls* – When set to "true", trunk group originated calls terminating to the network side include the trgp/trunk-context in the *Contact* header using the provisioned data from the Trunk Group Identity field. The tgrp value is taken from the Trunk Group Identity user and the trunk-context is taken from the Trunk Group Identity domain name. A pilot user is required for the trunk group when updating the Contact header this way.

3.8.5 X-Nortel-Profile Format

The Application Server can be configured to send the OTG in the *X-Nortel-Profile* header as:

X-Nortel-Profile = "X-Nortel-Profile" HCOLON 1*15(alphanum / "*" / "#" / "_" / ".")

Example message from the Application Server:

```
INVITE sip:+15146994609@192.168.8.76:5060; user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.8.76;branch=z9hG4bKBroadWorks.-1su2iau-192.168.8.76V50
From:"john10sipura south"<sip:5146994610@192.168.8.76;user=phone>;tag=598986802-117
To:<sip:+15146994609@192.168.8.76:5060;user=phone>
Call-ID:BW134004578230507-651953672@192.168.8.76
CSeq:486740914 INVITE
Contact:<sip:192.168.8.76:5060>
X-Nortel-Profile: 987
Remote-Party-ID:"john "<sip:5146994610@192.168.8.76;user=phone>;
    screen=yes;party=calling;privacy=off;id-type=subscriber
RPID-Privacy:party=calling;id-type=subscriber;privacy=off
Proxy-Require:privacy
Supported:100rel
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
Accept:multipart/mixed,application/media_control+xml,application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:421
```

If both the Originating Trunk Group (OTG) and the Destination Trunk Group (DTG) (see section *3.9 Destination Trunk Group Support*) are configured to use the X-Nortel-Profile format, there can be collision if the SIP 302 response from the Network Server contains both an OTG and a DTG value. In this situation, the Destination Trunk Group takes precedence and is sent in the *X-Nortel-Profile* header. The Originating Trunk Group value is discarded.

3.9 Destination Trunk Group Support

3.9.1 Origination

Cisco BroadWorks supports adding Destination Trunk Group (DTG) information to outgoing requests. DTG information provides additional detail on how to route a call. SIP network elements can use the DTG information to route the call based on some internal logic and/or service.

The Application Server can send the DTG information in two formats depending on the requirements of the network where Cisco BroadWorks is deployed. The two formats are “dtg” and “x-nortel-profile”, with “dtg” being the default format.

- If the parameter is set to “dtg”, the DTG information is passed in the outgoing INVITE request as a *dtg* parameter in the SIP URI added to the *Request-URI* and the *To* header.
- If the parameter is set to “x-nortel-profile”, an *X-Nortel-Profile* header is added to carry the DTG information in the outgoing SIP INVITE request.

The DTG allows interoperability with network elements that require Cisco BroadWorks to indicate a DTG for 0+ calls.

3.9.1.1 Network Server Configuration

The DTG value is configured at the Network Server. The Network Server selection is based on the caller and the dialed digits. All policies selecting a Routing NE support the new DTG selector:

- Destination Service Routing (DstSvcRtg)
- Equal Access (EqualAccess)
- Far-End Routing (FarEndRtg)
- Near-End Routing (NearEndRtg)

- Number Portability (NumberPortability)
- Physical Location (PhysLocation)
- Rate Center Based Routing (RCBasedRtg)
- Service Center Routing (SvcCtrRtg)
- Tandem Overflow (TandemOverflow)

Equal Access, Far-End Routing, Near-End Routing, and Tandem Overflow policies all based their routing on the RoutingNE routing list. This list supports a near-end and far-end DTG.

- The near-end DTG is returned when the originator is used to make the routing decision. This is true for the Equal Access, Near-End Routing, and Tandem Overflow policies.
- The far-end DTG is returned when the destination is used to make the routing decision. This is true for the Far-End Routing policy.

Once the Network Server has determined the appropriate DTG, it returns its value as part of the returned contacts in the 302 response to the Application Server.

The DTG is composed of 1 through 15 characters selected among:

[a..z][A..Z][0..9]*#-_.

Example response from the Network Server to the Application Server:

```
SIP/2.0 302 Moved temporarily
Via:SIP/2.0/UDP 192.168.8.76;branch=z9hG4bKBroadWorks.-1su2iau-192.168.12.60V5060-
0-486740867-1603108882-1179942004484-
From:"john10sipura south"<sip:+15146994610@192.168.8.76;user=phone>;tag=1603108882-
1179942004484-
To:<sip:6994609@192.168.12.60;user=phone>
Call-ID:BW134004484230507156506000 8@192.168.8.76
CSeq:486740867 INVITE
Contact<sip:+15146994609@192.168.8.76:5060;dtg=987;user=phone>;q=0.17;ct=OAP;ton=PU
BLIC;cat=INTERLAT
Content-Length:0
```

3.9.1.2 DTG Format

The Application Server can be configured to send the DTG in the *Request-URI* and the *To* header. The DTG appears as a *dtg* parameter of the SIP URI as follows.

```
uri-parameter = … / dtg-param
dtg-param = "dtg"  EQUAL dtg-value
dtg-value = trunk-group-label ["@" trunk-context descriptor ]
```

The syntax of the trunk-group-label and trunk-context descriptor is defined in *RFC 4904* [62]. The “@” sign is escaped in the *Contact* header URI as “%40”. The trunk-context descriptor is optional.

Example message from the Application Server:

```
INVITE sip:+15146994609@192.168.8.76:5060;dtg=987;user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.8.76;branch=z9hG4bKBroadWorks.-1su2iau-192.168.8.76V50
From:"john10sipura south"<sip:5146994610@192.168.8.76;user=phone>;tag=598986802-117
To:<sip:+15146994609@192.168.8.76:5060;dtg=987;user=phone>
Call-ID:BW134004578230507-651953672@192.168.8.76
CSeq:486740914 INVITE
Contact:<sip:192.168.8.76:5060>
Remote-Party-ID:"john"<sip:5146994610@192.168.8.76;user=phone>;
    screen=yes;party=calling;privacy=off;id-type=subscriber
RPID-Privacy:party=calling;id-type=subscriber;privacy=off
Proxy-Require:privacy
Supported:100rel
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
Accept:multipart/mixed,application/media_control+xml,application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:421
```

3.9.1.3 X-Nortel-Profile-Format

The Application Server can be configured to send the DTG in the *X-Nortel-Profile* header as:

X-Nortel-Profile = “X-Nortel-Profile” HCOLON 1*15(alphanum / "*" / "#" / "_" / ".")

Example message from the Application Server:

```
INVITE sip:+15146994609@192.168.8.76:5060; user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.8.76;branch=z9hG4bKBroadWorks.-1su2iau-192.168.8.76V50
From:"john10sipura south"<sip:5146994610@192.168.8.76;user=phone>;tag=598986802-117
To:<sip:+15146994609@192.168.8.76:5060;user=phone>
Call-ID:BW134004578230507-651953672@192.168.8.76
CSeq:486740914 INVITE
Contact:<sip:192.168.8.76:5060>
X-Nortel-Profile: 987
Remote-Party-ID:"john "<sip:5146994610@192.168.8.76;user=phone>;
    screen=yes;party=calling;privacy=off;id-type=subscriber
RPID-Privacy:party=calling;id-type=subscriber;privacy=off
Proxy-Require:privacy
Supported:100rel
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
Accept:multipart/mixed,application/media_control+xml,application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:421
```

If both the Originating Trunk Group (see section *3.8 Incoming/Outgoing OTG Support*) and the Destination Trunk Group are configured to use the X-Nortel-Profile format, there can be collision if the SIP 302 response from the Network Server contains both an OTG and a DTG value. In this situation, the Destination Trunk Group takes precedence and is sent in the *X-Nortel-Profile* header. The Originating Trunk Group value is discarded.

3.9.2 Termination

The Cisco BroadWorks Application Server can use destination trunk group information in the *Request-URI* of a SIP request when it attempts to identify a Cisco BroadWorks subscriber as the terminating party.

When the Application Server receives a SIP request from the network interface, it looks first for *tgrp* and *trunk-context* parameters in the *Request-URI*, in accordance with *RFC 4904*.

- If it finds these parameters, it looks for a matching *Trunk Group Identity* in its database and finds the corresponding trunk group.

- If the Application Server cannot find a matching *Trunk Group Identity*, then it ignores the *trgp* parameter and continues performing translations as though *tgrp* were not present.
- If it finds a match, then it checks the value of the *allowTerminationToTrunkIdentity* attribute of the trunk group.
  - If the value is “false”, then the Application Server ignores the *tgrp* parameter and continues performing translations as though *tgrp* were not present.
  - If the value is “true” and the trunk group has no pilot user, then the Application Server rejects the call with User Not Found treatment. Otherwise, it performs translations on the called number using the profile of the trunk group’s pilot user. This means, specifically, that the Application Server can match the extension of another Cisco BroadWorks subscriber in the same group or enterprise as the pilot user.

If the SIP request does not have a *tgrp* parameter in the *Request-URI* or if the tgrp parameter is ignored (as described in the preceding paragraph), then the Application Server looks for a *dtg* parameter in the *Request-URI*. If the *Request-URI* has a *dtg* parameter, the processing is similar to that described in the preceding paragraph. In this case, however, the Application Server looks for a matching *OTG/DTG Identity* to find a trunk group and checks the value of the trunk group’s *allowTerminationToDtgIdentity* attribute.

3.10 Cisco BroadWorks Support for Request History

3.10.1 Processing Model

Cisco BroadWorks supports the *History-Info* header and the *Diversion* header, as well as interworking between them.

When Cisco BroadWorks receives a *History-Info* or *Diversion* header, it extracts the information from each entry in the header into a common internal data representation. This internal representation allows Cisco BroadWorks to process the data without regard to its source. Likewise, when Cisco BroadWorks sends a *History-Info* or *Diversion* header, it uses the data in the internal representation to create the header content. This internal representation includes the following information, which is common to both the *History-Info* header and the *Diversion* header:

- Display Name
- URI
- Privacy Indicator
- Sequence Information (for example, information from *index* in the *History-Info* header)
- Diversion Reason
- Cisco BroadWorks proprietary parameters

Additionally, the internal representation contains preserved information that is source-specific. In general, Cisco BroadWorks does not use this source-specific information for call processing, but Cisco BroadWorks preserves the information so that it can copy the information to an outgoing header. For example, if Cisco BroadWorks receives a *History-Info* header with unrecognized parameters, it preserves these parameters so that it can send them in an outgoing *History-Info* header (or discard them if it sends an outgoing *Diversion* header).

In general, if Cisco BroadWorks receives an incoming *History-Info* header and sends an outgoing *History-Info* header, the conversion to an internal representation and back has little, if any, externally observable impact. Similarly, if Cisco BroadWorks receives an incoming *Diversion* header and sends an outgoing *Diversion* header, the conversion has little observable impact. However, this model of internal processing is useful for explaining how Cisco BroadWorks converts *History-Info* to *Diversion*, and vice versa. It also emphasizes the fact that for call processing purposes, Cisco BroadWorks treats *History-Info* and *Diversion* largely the same.

For the internal diversion reason, Cisco BroadWorks allows the values listed in the following table. When Cisco BroadWorks itself is responsible for a diversion, it adds a new internal entry that may have a proprietary value for the diversion reason. These proprietary values are identified in the following table. In general, Cisco BroadWorks sends these proprietary values only on the network interface and only when sending an INVITE request for a distributed group call (DGC). In all other situations, Cisco BroadWorks converts the proprietary values to non-proprietary values.

Internal Diversion Reason	Conformance	Service Usage (Not Exhaustive)
unknown	*RFC 5806*	Redirection Service, Group Paging Service, Instant Group Call Service
user-busy	*RFC 5806*	Call Forward Busy, Directory Number Hunting Agent Service, Trunk Group Service, Call Center Service, Series Completion Service
no-answer	*RFC 5806*	Call Forward No Answer, Directory Number Hunting Agent Service, Hunt Group Service, Executive Rollover Action
unavailable	*RFC 5806*	Call Center Agent Service, Voice Mail Service, Directory Number Hunting Agent Service, Hunt Group Service, Trunk Group Service, Call Forward No Reachable Terminator Service, Intercept Terminator Service
unconditional	*RFC 5806*	Call Forward Always, Group Night Forwarding Service, Directory Number Hunting Agent Service, Executive- Assistant Terminating Service
deflection	*RFC 5806*	Client Transfer, Blind Transfer, Find-me/Follow-me Push Service, Intercept Terminating Service, BroadWorks Anywhere Portal Service, Auto Attendant Service
time-of-day	*RFC 5806*	Group Night Forwarding Service, Service Control Function Service
do-not-disturb	*RFC 5806*	This reason is accepted from incoming redirection header.
follow-me	*RFC 5806*	BroadWorks Mobility Service, Simultaneous Ring Service, Sequential Ring Service, Shared Call Appearance Service, Remote Office Service, Find-me/Follow-me Service
out-of-service	*RFC 5806*	This reason is accepted from incoming redirection header.
away	*RFC 5806*	This reason is accepted from incoming redirection header.
transfer	Cisco BroadWorks proprietary	Client Call Control 2 Service

Internal Diversion Reason	Conformance	Service Usage (Not Exhaustive)
voicemail	Cisco BroadWorks proprietary	Client Call Control 2 Service
hunt-group	Cisco BroadWorks proprietary	Hunt Group Service
call-center	Cisco BroadWorks proprietary	Call Center Service
route-point	Cisco BroadWorks proprietary	Route Point Service
BW-ImplicitID	Cisco BroadWorks proprietary	Voice Portal Service, System Voice Portal Service
BW-ExplicitID	Cisco BroadWorks proprietary	Voice Portal Service, System Voice Portal Service

NOTE: The Application Server uses only the *Diversion* header when communicating with the Network Server.

3.10.2 Compliance

3.10.2.1 Support for History-Info (RFC 4244, RFC 7044)

The *History-Info* header is designed to capture the complete *Request-URI* history at every network element along the path from the User Agent Client (UAC) to the final User Agent Server (UAS). A proxy server or other intermediary that strictly conforms to *RFC 7044* adds a new *History-Info* header entry even if it does not change the *Request-URI*. This means the *History-Info* header can contain entries that record a call diversion, as well as entries that do not. An example of a non-diversion entry is an entry added by a network gateway to record the redirection following a query to the Cisco BroadWorks Network Server.

In practice, not all network elements use the *History-Info* header to record complete *Request-URI* history. Some network elements use the *History-Info* header as an alternative to the *Diversion* header to record only call diversions. Cisco BroadWorks is one such network element. Cisco BroadWorks adds a new *History-Info* header only when it needs to record a "diversion", loosely defined.

Recognizing a need to distinguish between diversion and non-diversion entries in the *History-Info* header, 3GPP codified the relevant criteria in *TS 24.604*, which is also backed in the IETF by *RFC 4458* and *RFC 6044*. Cisco BroadWorks supports these specifications.

If Cisco BroadWorks receives an initial INVITE request that contains a *History-Info* header and no *Diversion* header, it processes the *History-Info* header without requiring any special configuration. If the request also has a *Diversion* header, then Cisco BroadWorks may process the *History-Info* header or ignore it, depending on configuration.

If Cisco BroadWorks sends a SIP request with request history information (diversion entries or non-diversion entries), it adds a *History-Info* header if it is configured to do so. On the network interface, it sends a *History-Info* header if *useHistoryInfoOnNetworkSide* is set to "true". On the access interface, it sends a *History-Info* header if the access device has the *Use History Info Header* device option enabled.

Cisco BroadWorks is not fully compliant with *RFC 7044*. The following points summarize the Cisco BroadWorks implementation:

- Cisco BroadWorks does not support the "histinfo" option tag.
  - By default, Cisco BroadWorks does not relay "histinfo" from an incoming request to an outgoing request.
  - Cisco BroadWorks ignores "histinfo" in received requests – that is, there is no change to the Application Server behavior when "histinfo" is present.
  - Cisco BroadWorks does not add "histinfo" to outgoing requests.
- Cisco BroadWorks supports the *rc*, *mp*, or *np* parameters in syntax but not semantics.
  - Cisco BroadWorks recognizes *rc*, *mp*, *np*, which means Cisco BroadWorks does not treat them as other unrecognized parameters.
  - Cisco BroadWorks may copy these parameters from an incoming *History-Info* entry to an outgoing *History-Info* entry.
  - Cisco BroadWorks does not add these parameters to a new *History-Info* entry or to an entry that it received as a *Diversion* entry.
- Cisco BroadWorks uses "1" instead of "0" as the index value to fill gaps between *History-Info* entries. (This behavior is typical of implementations that support *RFC 4244* but not *RFC 7044*. Furthermore, *RFC 6044* recommends using "1".)
- Cisco BroadWorks supports the use of the *Reason* header embedded in the *History-Info* entry's URI.
- Cisco BroadWorks supports the use of the *Privacy* header embedded in the *History-Info* entry's URI. Cisco BroadWorks can act as a privacy service and anonymize entries in the *History-Info* header when it sends an outgoing request to an untrusted device.
- Cisco BroadWorks supports only a single branch of the request history. When Cisco BroadWorks processes a *History-Info* header from an incoming SIP request, it ignores all branches except the active branch (that is, the branch that contains the current *Request-URI*). When Cisco BroadWorks sends an outgoing SIP request, the *History-Info* header has only a single branch, even if the incoming request had multiple branches.
- Cisco BroadWorks does not directly copy the URI from an incoming *History-Info* entry to an outgoing *History-Info* entry. Cisco BroadWorks may rewrite the URI.
  - If the user part is a phone number, Cisco BroadWorks may change it to national format or to E.164 format.
  - Cisco BroadWorks may change the host part to the device endpoint owner's call processing domain or the system domain.
  - Cisco BroadWorks may change the entry for the current *Request-URI*, if necessary, to match current *Request-URI* (which may have changed from the received *Request-URI*).

Cisco BroadWorks may add proprietary parameters or parameter values to the *History-Info* header. The following ABNF defines the syntax Cisco BroadWorks supports. This syntax is fully compatible with the *RFC 7044* syntax (and *RFC 4244* syntax).

```
History-Info = "History-Info" HCOLON hi-entry *(COMMA hi-entry)

hi-entry = hi-targeted-to-uri *(SEMI hi-param)

hi-targeted-to-uri = name-addr

hi-param = hi-index / hi-target-param / hi-extension

hi-index = "index" EQUAL index-val

index-val =  number *("." number)

number =  [ %x31-39 *DIGIT ] DIGIT

hi-target-param = rc-param / mp-param / np-param

rc-param = "rc" EQUAL index-val

mp-param = "mp" EQUAL index-val

np-param = "np" EQUAL index-val

hi-extension = generic-param / bw-param / bw-network-param

; BroadWorks proprietary parameters
bw-param = family-param / diversion-inhibited-param /
    network-inhibited-param / answered-count-param /
    intercept-exempt-param

; BroadWorks proprietary parameters sent only on network interface
bw-network-param = answered-param / pbx-device-param /
    delay-ccm-param / simring-list-param / xfer-cc-param /
    x-bw-dnh-param / x-bw-release-redirection-param /
    user-id-param / x-bw-vm-transfer-user-id-param /
    hg-cc-param / fax-deposit-param / external-vm-deposit-param /
    x-bw-find-me-follow-me-param / x-bw-fmfm-call-push-param /
    x-bw-phone-list-name-param / x-bw-igc-param

family-param = "family"

diversion-inhibited-param = "diversion-inhibited"

network-inhibited-param = "network-inhibited"

answered-count-param = "answered-count" EQUALS 1*DIGIT

intercept-exempt-param = "intercept-exempt"

answered-param = "answered"

pbx-device-param = "pbxDevice"

delay-ccm-param = "delay-ccm"

simring-list-param = "simring-list" EQUALS quoted-string

xfer-cc-param = "xferCC"
```

```
x-bw-dnh-param = "x-bw-dnh"

x-bw-release-redirection-param = "x-bw-release-redirection"

user-id-param = "user-id" EQUALS quoted-string

x-bw-vm-transfer-user-id-param = "x-bw-vm-transfer-user-id" EQUALS
    quoted-string

hg-cc-param = "hg-cc"

fax-deposit-param = "fax-deposit"

external-vm-deposit-param = "external-vm-deposit"

x-bw-find-me-follow-me-param = "x-bw-find-me-follow-me"

x-bw-fmfm-call-push-param = "x-bw-fmfm-call-push"

x-bw-phone-list-name-param = "x-bw-phone-list-name" EQUALS quoted-string

x-bw-igc-param = "x-bw-igc"
```

3.10.2.2 Support for Diversion (RFC 5806)

Cisco BroadWorks supports the *Diversion* header as defined in *RFC 5806*.

If Cisco BroadWorks receives an initial INVITE request that contains a *Diversion* header and no *History-Info* header, it processes the *Diversion* header without requiring any special configuration. If the request also has a *History-Info* header, then Cisco BroadWorks may process the *Diversion* header or ignore it, depending on configuration.

If Cisco BroadWorks sends a SIP request with diversion information, it adds a *Diversion* header if it is configured to do so. On the network interface, it sends a *Diversion* header if *useHistoryInfoOnNetworkSide* is set to "false". On the access interface, it sends a *Diversion* header if the access device has the *Use History Info Header* device option disabled.

The following points summarize the Cisco BroadWorks implementation:

- Cisco BroadWorks can parse the *limit* parameter, but Cisco BroadWorks does not otherwise support this parameter.
- Cisco BroadWorks can parse the *screen* parameter, but Cisco BroadWorks does not otherwise support this parameter.
- Cisco BroadWorks may add proprietary parameters to the *Diversion* header, particularly when Cisco BroadWorks itself is responsible for a call diversion.
- Cisco BroadWorks may add proprietary values to the *reason* parameter, particularly when Cisco BroadWorks itself is responsible for a call diversion. Cisco BroadWorks only adds these proprietary reasons when sending an INVITE request on the network interface and only in an INVITE request for a distributed group call (DGC).

The following ABNF defines the *Diversion* header syntax Cisco BroadWorks supports. This syntax is fully compatible with the *RFC 5806* syntax.

```
Diversion = "Diversion" HCOLON diversion-params
    *(COMMA diversion-params)

diversion-params = name-addr *(SEMI (diversion-reason /
```

```
    diversion-counter / diversion-limit / diversion-privacy /
    diversion-screen / bw-param / bw-network-param /
    diversion-extension))

diversion-reason = "reason" EQUAL ("unknown" / "user-busy" /
    "no-answer" / "unavailable" / "unconditional" / "time-of-day" /
    "do-not-disturb" / "deflection" / "follow-me" / "out-of-service" /
    "away" / bw-reason / token / quoted-string)

diversion-counter = "counter" EQUAL 1*2DIGIT

diversion-limit = "limit" EQUAL 1*2DIGIT

diversion-privacy = "privacy" EQUAL ("full" / "name" / "uri" / "off" /
    token / quoted-string)

diversion-screen = "screen" EQUAL ("yes" / "no" / token / quoted-string)

diversion-extension = token [EQUAL (token / quoted-string)]

; BroadWorks proprietary reason values
bw-reason = "transfer" / "voicemail" / "BW-ImplicitID" /
    "BW-ExplicitID" / "hunt-group" / "call-center" / "route-point"

; BroadWorks proprietary parameters
bw-param = family-param / diversion-inhibited-param /
    network-inhibited-param / answered-count-param /
    intercept-exempt-param

; BroadWorks proprietary parameters sent only on network interface
bw-network-param = answered-param / pbx-device-param /
    delay-ccm-param / simring-list-param / xfer-cc-param /
    x-bw-dnh-param / x-bw-release-redirection-param /
    user-id-param / x-bw-vm-transfer-user-id-param /
    hg-cc-param / fax-deposit-param / external-vm-deposit-param /
    x-bw-find-me-follow-me-param / x-bw-fmfm-call-push-param /
    x-bw-phone-list-name-param / x-bw-igc-param

family-param = "family"

diversion-inhibited-param = "diversion-inhibited"

network-inhibited-param = "network-inhibited"

answered-count-param = "answered-count" EQUALS 1*DIGIT

intercept-exempt-param = "intercept-exempt"

answered-param = "answered"

pbx-device-param = "pbxDevice"

delay-ccm-param = "delay-ccm"

simring-list-param = "simring-list" EQUALS quoted-string

xfer-cc-param = "xferCC"

x-bw-dnh-param = "x-bw-dnh"

x-bw-release-redirection-param = "x-bw-release-redirection"
```

```
user-id-param = "user-id" EQUALS quoted-string

x-bw-vm-transfer-user-id-param = "x-bw-vm-transfer-user-id" EQUALS
    quoted-string

hg-cc-param = "hg-cc"

fax-deposit-param = "fax-deposit"

external-vm-deposit-param = "external-vm-deposit"

x-bw-find-me-follow-me-param = "x-bw-find-me-follow-me"

x-bw-fmfm-call-push-param = "x-bw-fmfm-call-push"

x-bw-phone-list-name-param = "x-bw-phone-list-name" EQUALS quoted-string

x-bw-igc-param = "x-bw-igc"
```

3.10.2.3 History-Info and Diversion Interworking (RFC 6044)

RFC 6044 describes how a SIP server should support interworking between the *History-Info* and *Diversion* headers. Cisco BroadWorks can be configured to follow *RFC 6044* recommendations. This conforming behavior is enabled via two settings:

- The SIP parameter *enableRFC6044* must be set to “true”.
- And, support for the *cause* parameter (*RFC 4458*) must be enabled.
  - On the network interface, the SIP parameter *supportCauseParameter* must be set to “true”.
  - On the access interface, the access device must have the *Support Cause Parameter* device option enabled.

Implementation of *RFC 6044* depends on an awareness of the headers supported by the originating endpoint and the terminating endpoint. When Cisco BroadWorks receives an INVITE request that has only a *History-Info* header, it infers that the originating endpoint supports *History-Info*. Likewise, if Cisco BroadWorks receives an INVITE request that has only a *Diversion* header, it infers that the originating endpoint supports *Diversion*. When Cisco BroadWorks sends an INVITE request, however, it does not make such an inference, but depends on configuration. If the SIP parameter *useHistoryInfoOnNetworkSide* is set to “true”, then it assumes all network endpoints support *History-Info*; otherwise, it assumes all network endpoints support *Diversion*. If an access device has the option *Use History Info Header* enabled, then Cisco BroadWorks assumes the access device supports *History-Info*; otherwise, it assumes the access device supports *Diversion*.

3.10.3 Receiving Request History

3.10.3.1 Receiving Both the History-Info Header and the Diversion Header

If Cisco BroadWorks receives a request that contains both a *History-Info* header and a *Diversion* header, then it may process only one header, or both headers, depending on its configuration. If *RFC 6044* conformance is enabled, then it processes both headers, as described below. Otherwise, it processes only the *History-Info* header or only the *Diversion* header, based on the SIP parameter *redirectionHeaderPriority*. If *redirectionHeaderPriority* is set to "historyInfo", then Cisco BroadWorks ignores the *Diversion* header and processes the *History-Info* header as described in section *3.10.3.3 Receiving the History-Info Header*. Alternatively, if *redirectionHeaderPriority* is set to "diversion", then Cisco BroadWorks ignores the *History-Info* header and processes the *Diversion* header as described in section *3.10.3.4 Receiving the Diversion Header*. The *redirectionHeaderPriority* default value is "historyInfo".

If *RFC 6044* conformance is enabled, then Cisco BroadWorks merges the *History-Info* header entries and the *Diversion* header entries. Since the request has a *History-Info* header and a *Diversion* header, Cisco BroadWorks assumes that the adjacent upstream network element supports *Diversion* and at least one other upstream network element supports *History-Info*. *RFC 6044* discusses this scenario, and Cisco BroadWorks follows the *RFC 6044* procedure to merge the information in the *History-Info* and *Diversion* headers.

When merging *History-Info* headers and *Diversion* headers, Cisco BroadWorks first accepts all *History-Info* headers. Next, it checks each *Diversion* header to see if it is a duplicate of a *History-Info* header. If the *Diversion* header is a duplicate, Cisco BroadWorks discards it; if not, then Cisco BroadWorks accepts it. Cisco BroadWorks assumes that any non-duplicate *Diversion* header entries are more recent that the most recent *History-Info* header.

To determine if a *Diversion* header entry is a duplicate of a *History-Info* header entry, Cisco BroadWorks compares the SIP URIs of the two entries and asserts a match if the URI user parts match and the URI host parts match.

3.10.3.2 History-Info and Diversion Header Screening

For both the *History-Info* header and the *Diversion* header, Cisco BroadWorks screens all received header entries, and it may remove entries that it considers untrusted. Cisco BroadWorks accepts the header entry if one of the following conditions is true:

- Cisco BroadWorks received the SIP request from the network interface.
- Cisco BroadWorks received the SIP request from a trusted access device.
- Cisco BroadWorks determines that the request is for a PBX redirection.
- Cisco BroadWorks received the SIP request from an untrusted access device, but these additional conditions are satisfied: (a) the user part of the URI is a phone number, (b) the host part of the URI matches an address associated with the access device. This address could be the host part of the user's line/port or the address provisioned for the access device.

If none of these conditions are satisfied, then Cisco BroadWorks removes the header entry.

3.10.3.3 Receiving the History-Info Header

When Cisco BroadWorks begins processing the *History-Info* header, it screens all *History-Info* header entries in order to prevent spoofing. This screening procedure is described in section *3.10.3.2 History-Info and Diversion Header Screening*.

After screening, Cisco BroadWorks divides the accepted entries into diversion entries and non-diversion entries. Diversion entries are the entries that record a call diversion; non-diversion entries are all other entries. If Cisco BroadWorks has *RFC 6044* conformance disabled, then it considers all *History-Info* header entries to be diversion entries, except for the entry that records the current *Request-URI*. However, if *RFC 6044* conformance is enabled, then Cisco BroadWorks applies the following criteria to decide whether an entry is a diversion entry or non-diversion entry:

- If the entry is the most recent entry and it records the current *Request-URI*, then the entry is a non-diversion entry,
- Else, if the entry that follows this entry has a *cause* parameter in its URI, then this entry is a diversion entry,
- Else, if the entry has a *Reason* header embedded in its URI, then it is a diversion entry,
- Else, the entry is a non-diversion entry.

As Cisco BroadWorks processes the *History-Info* header entries, it creates an internal representation entry for every diversion entry in a diversion entries list and an internal representation entry for every non-diversion entry in a non-diversion entries list.

The following table shows how Cisco BroadWorks sets the values of its internal representation from the syntax elements of the *History-Info* header entry.

Internal Representation	Source
Display Name	Display name from the *hi-targeted-to* element.
URI	URI from the *hi-targeted-to* element. If the user part is a phone number, Cisco BroadWorks changes the host part of the URI to the endpoint owner's call processing domain or the system domain. In IMS mode, Cisco BroadWorks changes the host part only if it received the INVITE request from the access interface.
Privacy Indicator	If the INVITE request has a *Privacy* header with one of the values "history", "header", or "session", then Cisco BroadWorks sets the Privacy Indicator to "anonymous". Else, if the entry's URI has an embedded *Privacy* header with the value "history" header, then Cisco BroadWorks sets the Privacy Indicator to "anonymous". Else, Cisco BroadWorks sets the Privacy Indicator to "public".
Sequence Information	Value of the *index* parameter (converted to an internal representation).
Diversion Reason	Value determined by the URI-embedded *Reason* header or the *cause* parameter. See the separate explanation.

If the *History-Info* entry's URI has an embedded *Reason* header with a Diversion protocol entry, then Cisco BroadWorks sets the diversion reason from this *Reason* header entry. Otherwise, if the SIP parameter *supportCauseParameter* is enabled, then it sets the diversion reason by translating the *cause* parameter from the following entry.

If Cisco BroadWorks gets the diversion reason from the embedded *Reason* header, then it uses that value directly. However, when Cisco BroadWorks gets the diversion reason from the *cause* parameter of the following entry, it performs a lookup into a configurable table. The following table provides the default values for the lookup table.

Cause Value	Internal Diversion Reason
404	*unknown*
486	*user-busy*
408	*no-answer*
503	*unavailable*
302	*unconditional*
480	*deflection*
(other)	*unknown*

A system administrator may configure the lookup table from the CLI at the */Interface/SIP/DiversionReasonMap* level. By default, this table has entries that conform to the *RFC 6044* recommendations.

3.10.3.4 Receiving the Diversion Header

When Cisco BroadWorks begins processing the *Diversion* header, it screens all *Diversion* header entries in order to prevent spoofing. This screening procedure is described in section *3.10.3.2 History-Info and Diversion Header Screening*.

Cisco BroadWorks processes all the entries in the *Diversion* header, adding an internal representation entry for each entry in the *Diversion* header. The following table shows how Cisco BroadWorks sets the values of its internal representation from the syntax elements of the *Diversion* header entry.

Internal Representation	Syntax Element
Display Name	Display name part of the *name-addr* element.
URI	URI part of the *name-addr* element. If the user part is a phone number, Cisco BroadWorks changes the host part of the URI to the endpoint owner's call processing domain or the system domain. In IMS mode, Cisco BroadWorks changes the host part only if it received the INVITE request from the access interface.

Internal Representation	Syntax Element
Privacy Indicator	If the INVITE request has a *Privacy* header with one of the values “history”, “header”, or “session”, then Cisco BroadWorks sets the Privacy Indicator to “anonymous”. Otherwise, Cisco BroadWorks derives the Privacy Indicator from the value of the *privacy* parameter as follows: ▪ If the value is “full”, then Cisco BroadWorks sets the Privacy Indicator to “anonymous”. ▪ If the value is “name”, then Cisco BroadWorks sets the Privacy Indicator to “anonymous-name”. ▪ If the value is “uri”, then Cisco BroadWorks sets the Privacy Indicator to “anonymous-uri”. ▪ If the value is “off” or the *privacy* parameter is omitted, then Cisco BroadWorks sets the Privacy Indicator to “public”.
Sequence Information	Entry position and value of the *counter* parameter.
Diversion Reason	Value of the *reason* parameter.

3.10.4 Sending Request History

When Cisco BroadWorks has request history information to send in an outgoing request, it sends a *History-Info* header, a *Diversion* header, or both, depending on configuration and scenario conditions.

Cisco BroadWorks sends both headers only if all of the following conditions are satisfied:

- *RFC 6044* conformance is enabled.
- By configuration, Cisco BroadWorks knows that the destination endpoint expects the *Diversion* header instead of the *History-Info* header. For a network endpoint, this means the SIP parameter *useHistoryInfoOnNetworkSide* is set to “false”. For an access device endpoint, this means the access device has the *Use History Info Header* device option disabled.
- Following *RFC 6044* recommendations, Cisco BroadWorks must send a *History-Info* header in addition to the *Diversion* header so that no request history information is lost. The following points list two scenarios where this condition is true:
  - Cisco BroadWorks received an incoming *History-Info* header with entries that could not be converted to *Diversion* without loss of information. For example, one or more *History-Info* entries contain parameters that Cisco BroadWorks does not recognize.
  - Cisco BroadWorks has non-diversion entries (which originated as received *History-Info* entries), which it cannot convert to *Diversion* entries.

When Cisco BroadWorks sends a *History-Info* header and a *Diversion* header, it only adds *History-Info* header entries for those diversion or non-diversion internal entries that originated as received *History-Info* entries. If Cisco BroadWorks added new internal entries for Cisco BroadWorks service-related call diversions, then those entries appear only as *Diversion* entries in the outgoing SIP request.

If the conditions for sending both headers are not satisfied, the Cisco BroadWorks sends only a *History-Info* header or only a *Diversion* header, depending on configuration. On the network interface, Cisco BroadWorks sends the *History-Info* header if the SIP parameter *useHistoryInfoOnNetworkSide* is set to "true" and sends the *Diversion* header otherwise. On the access interface, Cisco BroadWorks sends the *History-Info* header if the access device has the *Use History Info Header* device option enabled and sends the *Diversion* header otherwise.

3.10.4.1 Sending the History-Info Header

When Cisco BroadWorks has request history information to send in an outgoing request, it sends the *History-Info* header if it is configured to do so. On the network interface, Cisco BroadWorks sends the *History-Info* header if the SIP parameter *useHistoryInfoOnNetworkSide* is set to "true". On the access interface, Cisco BroadWorks sends the *History-Info* header if the access device has the *Use History Info Header* device option enabled.

Cisco BroadWorks creates a *History-Info* entry in the outgoing request for each entry in its internal diversion entry list and its internal non-diversion entry list. The following points describe how Cisco BroadWorks processes the internal representation to generate the *History-Info* entry:

- Display Name – If privacy protection is required, Cisco BroadWorks anonymizes the display name. Otherwise, Cisco BroadWorks uses the display name from the internal representation directly.

  Cisco BroadWorks decides that privacy protection is required if the destination endpoint is not trusted and the Privacy Indicator is set to "anonymous" or "anonymous-name".

  To make the display name anonymous, Cisco BroadWorks sets it to the value of the SIP parameter *restrictedDisplayName*, which has the default value "Anonymous".

- URI – If privacy protection is required, Cisco BroadWorks anonymizes the URI. Otherwise, Cisco BroadWorks uses the URI from the internal representation, possibly converting the user part of the URI to E.164 format or national format if it is a phone number.

  Cisco BroadWorks decides that privacy protection is required if the destination endpoint is not trusted and the Privacy Indicator is set to "anonymous" or "anonymous-uri".

  To make the URI anonymous, Cisco BroadWorks sets it to "sip:anonymous@anonymous.invalid".

- Index – Cisco BroadWorks generates the value of the *index* parameter using the sequence information of the internal representation.

  If the entry originated as a received *History-Info* entry, then Cisco BroadWorks uses the received *index* value.

  If the entry originated as a received *Diversion* entry, then Cisco BroadWorks uses the value of the *Diversion* entry's *counter* parameter to generate the *index* value. If the *counter* value is "1" or if there is no *counter* value, then Cisco BroadWorks adds a new *index* level with the value "1". If the *counter* value is greater than 1, then Cisco BroadWorks adds enough additional levels to cover the missing diversion entries. For these additional levels, Cisco BroadWorks uses the value "1".

Examples:

If the *index* value of the preceding entry is “1.2” and the *counter* value is “1”, then Cisco BroadWorks generates the *index* value “1.2.1”.

If the *index* value of the preceding entry is “1.2” and the *counter* value is “3”, then Cisco BroadWorks generates the *index* value “1.2.1.1.1”.

If Cisco BroadWorks itself added the entry to record a new diversion, then Cisco BroadWorks adds a new *index* level with the value “1”.

- *Privacy* header in URI – If privacy protection is required and Cisco BroadWorks is sending the SIP request to a trusted endpoint, then it adds a URI-embedded *Privacy* header with the value “history”. However, if the SIP request has a *Privacy* header with the value “history”, then Cisco BroadWorks omits the URI-embedded *Privacy* header (which would be redundant).

  Cisco BroadWorks decides that privacy protection is required if the privacy indicator is set to “anonymous”, “anonymous-name”, or “anonymous-uri”.

- *Reason* header in URI – Cisco BroadWorks may add a URI-embedded *Reason* header with SIP entry or a Diversion entry. See the discussion that follows.

- *cause* URI parameter – If the entry originated as a received *History-Info* entry that had an associated *cause* value, then Cisco BroadWorks uses that *cause* value. (Note that the “associated” *cause* value is the value of the *cause* URI parameter of the following entry.) Otherwise, if Cisco BroadWorks is configured to support the *cause* parameter, then it converts the diversion reason to a *cause* value. See the discussion that follows for details on the conversion. As required by *RFC 6044*, Cisco BroadWorks adds this *cause* value to the URI of the *History-Info* entry that follows the current one.

- Other URI-embedded headers – If *RFC 6044* conformance is enabled and the entry originated as a received *History-Info* entry, Cisco BroadWorks copies any other URI-embedded headers (other than *Privacy* and *Reason*, which Cisco BroadWorks processes independently).

- Other URI parameters – If *RFC 6044* conformance is enabled and the entry originated as a received *History-Info* entry, then Cisco BroadWorks copies any other URI parameters (other than *cause*, which Cisco BroadWorks processes independently).

- Other header field parameters – If *RFC 6044* conformance is enabled and the entry originated as a received *History-Info* entry, then Cisco BroadWorks copies any other parameters in the *History-Info* entry. This includes the *rc*, *mp*, and *np* parameters.

- Cisco BroadWorks proprietary parameters – If Cisco BroadWorks created a new entry for a Cisco BroadWorks service-related diversion, then it may add proprietary parameters.

Depending on configuration, as well as the scenario conditions, Cisco BroadWorks may add a URI-embedded *Reason* header.

- If the entry originated as a *History-Info* entry that contained a URI-embedded *Reason* header, then Cisco BroadWorks copies this *Reason* header to the outgoing *History-Info* entry.

- If *cause* parameter support is disabled, then Cisco BroadWorks adds the *Reason* header to record the diversion reason. Depending on the value of the diversion reason, Cisco BroadWorks may add an entry for the Diversion protocol and possibly an entry for the SIP protocol.

Examples:

The diversion reason is “user-busy”. Cisco BroadWorks adds a *Reason* header entry for both the SIP protocol and the Diversion protocol. The *Reason* header is shown as follows.

Reason: SIP;cause=486;text="Busy Here",Diversion;text="user-busy"

The diversion reason is “away”. Cisco BroadWorks adds a *Reason* header entry for only the Diversion protocol. The *Reason* header is shown as follows.

Reason: Diversion;text="away"

When Cisco BroadWorks adds the *Reason* header to the URI, it encodes it with the URI encoding. The following is an example of a URI with an embedded *Reason* header:

sip:+12145550000@pstn.example?Reason=Diversion%3Btext%3D%22user-busy%22

If Cisco BroadWorks is configured to support the *cause* parameter and it does not have a received *cause* value to use for the outgoing *History-Info* entry, then it generates a value from the internal diversion reason using a lookup into a configurable table. A system administrator may change the table via the CLI at the */Interface/SIP/DiversionReasonMap* level. The following table provides the default entries for the lookup table. These default entries conform to *RFC 6044*.

Internal Diversion Reason	Cause Value
unknown	404
user-busy	486
no-answer	408
unavailable	503
unconditional	302
deflection	480
time-of-day	404
do-not-disturb	404
follow-me	404
out-of-service	404
away	404
transfer	404
voicemail	404
hunt-group	404
call-center	404
route-point	404
BW-ImplicitID	404
BW-ExplicitID	404

If Cisco BroadWorks has *RFC 6044* conformance enabled, then it adds *History-Info* entries to fill any gaps in the index values. For each such entry, Cisco BroadWorks sets the URI to "sip:anonymous@anonymous.invalid" and creates the *index* value by adding a new level with the sequence number set to "1". Furthermore, Cisco BroadWorks associates a cause of 404 with the entry, which means it adds a *cause* URI parameter with value 404 to the following *History-Info* entry.

Example:

Cisco BroadWorks has an internal diversion entry with a counter value of "3", which originated from the following *Diversion* entry:

<sip:+12145550000@pstn.example>;reason=busy;counter=3

Therefore, Cisco BroadWorks must add two entries to fill the gap. Cisco BroadWorks creates the four *History-Info* entries below. The first two entries are new entries Cisco BroadWorks added to fill the gap. The third entry directly corresponds to the received *Diversion* entry. The last entry corresponds to the current *Request-URI*.

<sip:anonymous@anonymous.invalid>;index=1,

<sip:anonymous@anonymous.invalid;cause=404>;index=1.1,

<sip:+12145550000@broadworks.net;cause=404>;index=1.1.1,

<sip:+19725550100@broadworks.net;cause=486>;index=1.1.1.1

If Cisco BroadWorks does not have *RFC 6044* conformance enabled, then it does not add new entries, but it does set the *index* value in such a way as to indicate a gap exists. The following shows how Cisco BroadWorks forms the *History-Info* header when *RFC 6044* conformance is disabled. Note that Cisco BroadWorks adds a URI-embedded *Reason* header instead of a *cause* URI parameter.

<sip:+12145550000@broadworks.net?Reason=SIP%3Bcause=486%3Btext%3D%22Busy%20Here%22%2CDiversion%3Btext%3D%22user-busy%22>;index=1.1.1,

<sip:+19725550100@broadworks.net>;index=1.1.1.1

3.10.4.2 Sending the Diversion Header

When Cisco BroadWorks has request history information to send in an outgoing request, it sends the *Diversion* header if it's configured to do so. On the network interface, Cisco BroadWorks sends the *Diversion* header if the SIP parameter *useHistoryInfoOnNetworkSide* is set to "false". On the access interface, Cisco BroadWorks sends the *Diversion* header if the access device has the *Use History Info Header* device option disabled.

Cisco BroadWorks creates a *Diversion* entry in the outgoing request for each entry in its internal diversion entry list. Cisco BroadWorks does not add any *Diversion* entries for its internal non-diversion entry list. The following points describe how Cisco BroadWorks processes the internal representation to generate the *Diversion* entry.

- Display Name – If privacy protection is required, Cisco BroadWorks anonymizes the display name. Otherwise, Cisco BroadWorks uses the display name from the internal representation directly.

  Cisco BroadWorks decides that privacy protection is required if the destination endpoint is not trusted and the Privacy Indicator is set to "anonymous" or "anonymous-name".

  To make the display name anonymous, Cisco BroadWorks sets it to the value of the SIP parameter *restrictedDisplayName*, which has the default value "Anonymous".

- URI – If privacy protection is required, Cisco BroadWorks anonymizes the URI. Otherwise, Cisco BroadWorks uses the URI from the internal representation, possibly converting the user part of the URI to E.164 format or national format.

  Cisco BroadWorks decides that privacy protection is required if the destination endpoint is not trusted and the Privacy Indicator is set to "anonymous" or "anonymous-uri".

  To make the URI anonymous, Cisco BroadWorks sets it to "sip:anonymous@anonymous.invalid".

- *counter* – If the entry originated as a *Diversion* entry, then Cisco BroadWorks copies the received *counter* value.

  If the entry originated as a *History-Info* entry, then Cisco BroadWorks determines the *counter* value by converting the *index* values of the received *History-Info* entries.

  If Cisco BroadWorks itself added the diversion entry, then it sets the *counter* value to "1".

- *reason* – If Cisco BroadWorks is sending the SIP request for a DGC leg, then it uses the value of the internal diversion reason directly. Otherwise, Cisco BroadWorks uses the value of the internal diversion reason if it is allowed by *RFC 5806* or uses "unknown" if the internal diversion reason is a proprietary reason.

- *privacy* – If Cisco BroadWorks is sending the SIP request to a trusted endpoint, then it adds a *privacy* parameter with the value set as follows:

  - If the privacy indicator is "anonymous", then the parameter value is "full".
  - If the privacy indicator is "anonymous-name", then the parameter value is "name".
  - If the privacy indicator is "anonymous-uri", then the parameter value is "uri".
  - If the privacy indicator is "public", then the parameter value is "off".

- Cisco BroadWorks proprietary parameters – If Cisco BroadWorks created a new entry for a Cisco BroadWorks service-related diversion, then it may add proprietary parameters.

3.10.5 History-Info Header in SIP Responses

Cisco BroadWorks can send the *History-Info* header in the *200* response to an initial INVITE request. To enable this behavior, a system administrator must set the SIP parameter *includeHistoryInfoInResponse* to "true". The default value for this parameter is "false".

If *includeHistoryInfoInResponse* is set to "true", then Cisco BroadWorks performs as follows:

- When Cisco BroadWorks receives an initial INVITE request, it saves the received *History-Info* header into a cache.
- When Cisco BroadWorks receives a *200* response to an initial INVITE request, if the endpoint that sent the response is trusted, Cisco BroadWorks accepts the *History-Info* header in that response. Alternatively, if the endpoint is untrusted, Cisco BroadWorks ignores the *History-Info* header.
- When Cisco BroadWorks sends an outgoing *200* response to an initial INVITE request, then it performs one of these alternatives:

- If it has received an incoming *200* response with an accepted *History-Info* header, then it copies this header from the incoming *200* response to the outgoing *200* response.
- Else, if it has a *History-Info* header stored in its cache, then it sends that *History-Info* header in the outgoing *200* response.
- Else, it does not send a *History-Info* header in the outgoing *200* response.

3.10.6 Diversion Inhibitor Signaling

Cisco BroadWorks supports the ability to inhibit call diversions on other call control platforms. Typically, this capability is used in a Class 5 switch overlay where Cisco BroadWorks provides the call control services. The Class 5 is configured to forward to Cisco BroadWorks for terminating call services. As such, when Cisco BroadWorks sends the call to the Class 5 for termination to the subscriber, it inhibits the call forwarding capability on the Class 5 to allow the call to terminate directly to the subscriber's phone rather than being forwarded back to Cisco BroadWorks. To enable this capability, an administrator must enable the device option *Forwarding Override*.

Cisco BroadWorks may also signal a diversion inhibited condition when requested by a FAC or when needed by a service such as Call Center.

3.10.6.1 Diversion Header

Cisco BroadWorks inhibits call diversions by including a *Diversion* entry in the outgoing INVITE request with the forwarding counter set to the configurable *defaultMaxRedirectionDepth* value in the */SubscriberMgmt/Policy/CallProcessing/CallLimits* level. This *Diversion* entry allows Cisco BroadWorks to override the diverting services in the remote call control platform, by exceeding the maximum number of call diversions allowed in the network. This causes most Class 5 switches and private branch exchanges (PBXs) to terminate the call directly to the phone associated with the called number rather than redirecting the call, which would occur if the *Diversion* entry were not included. Note that if more than one *Diversion* header entry exists in the outgoing INVITE request, then Cisco BroadWorks adds a new entry and the total redirection count of all *Diversion* header entries may exceed the configured *defaultMaxRedirectionDepth* value.

Following is an example of the *Diversion* entry added when Cisco BroadWorks inhibits call diversions.

```
Diversion:"John Doe"<sip:+12403645291@broadsoft.com>;reason=unknown;counter=6;
privacy=off
```

Cisco BroadWorks (optionally) adds a *Diversion* entry with the parameter *diversion-inhibited* and the *counter* set to the configurable *defaultMaxRedirectionDepth* value when "diversion inhibited" needs to be signaled outside the scope of Cisco BroadWorks. This occurs when Cisco BroadWorks sends an INVITE request for which diversion is inhibited, either through FAC dialing or implicitly for Hunt Group and Call Center redirections. The insertion of this *Diversion* entry parameter is configurable. The presence of the *diversion-inhibited* parameter in a *Diversion* entry indicates that this *Diversion* entry is an extra entry, which is significant because the diversion should be inhibited on this call.

```
Diversion:"John Doe"<sip:+12403645291@broadsoft.com>;reason=unknown;counter=6;
privacy=off;diversion-inhibited
```

Note that the two formats of inhibitions typically occur in different scenarios. Cisco BroadWorks uses the Class 5 overlay format when it sends an INVITE request to a Class 5 switch (or any other device) configured as the user device with the associated device option. Cisco BroadWorks uses the *diversion-inhibited* format when it sends an INVITE request for which diversion is inhibited, either through FAC dialing or implicitly for Hunt Group and Call Center redirection. If both apply simultaneously, then the *diversion-inhibited* format takes precedence.

3.10.6.2 History-Info Header

When the diversion is inhibited or the *Forwarding* Override access device option is enabled, Cisco BroadWorks modifies the *Request-URI* entry of the *History-Info* header using the following rules:

- Add the *diversion-inhibited* parameter to the *Request-URI* entry.
- If the *History-Info* header has at least one valid entry (this is a redirection), the number of levels in the *Request-URI* index must be equal to *defaultMaxRedirectionDepth + 1.*
- If the *History-Info* header is absent or empty (this is an origination), the number of index levels is equal to the *defaultMaxRedirectionDepth + 1* and all levels are set to "1".

3.10.7 Call Flows

3.10.7.1 History-Info Header to History-Info Header with Cisco BroadWorks Call Forward Always

Figure 2 shows message flow for an incoming *History-Info* header within the INVITE request. Cisco BroadWorks receives the request with a *History-Info* header having sub-branch entries (1.1 and 1.2).

Cisco BroadWorks preserves only the active branch of the *History-Info* header. All *History-Info* entries in parallel branches are discarded (not proxied) by Cisco BroadWorks. The index received by Cisco BroadWorks is preserved and incremented for the subsequent redirections.

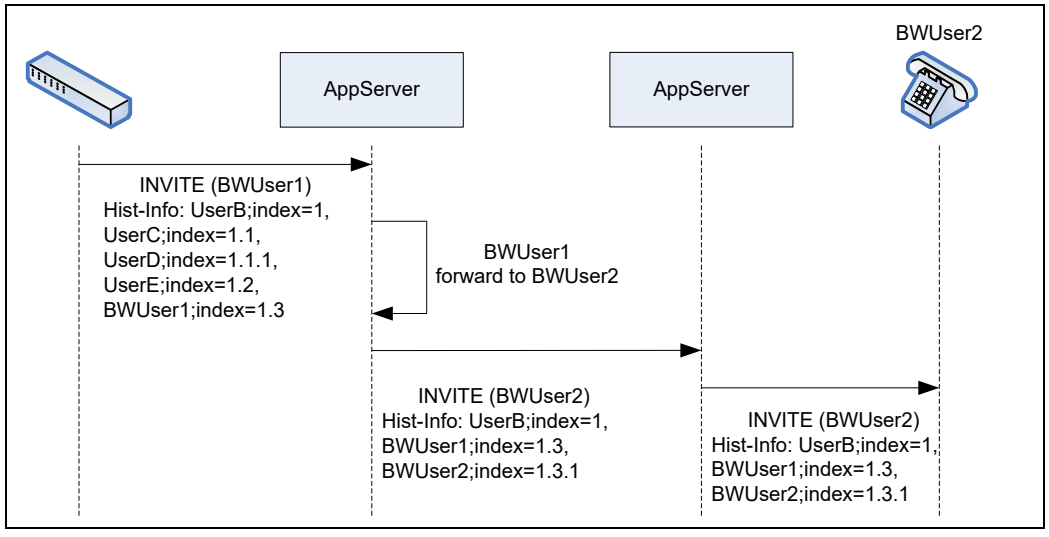

Figure 2 History-Info Indexing

3.10.7.2 History-Info Header with Diversion Inhibitor Signaling

When the diversion is inhibited or the *Forwarding Override* access device option is enabled, Cisco BroadWorks modifies the *Request-URI* entry of the *History-Info* header using the following rules:

1) Add the *diversion-inhibited* parameter to the *Request-URI* entry.

2) If the *History-Info* header has at least one valid entry (this is a redirection), the number of levels in the *Request-URI* index must be equal to *defaultMaxRedirectionDepth + 1.*

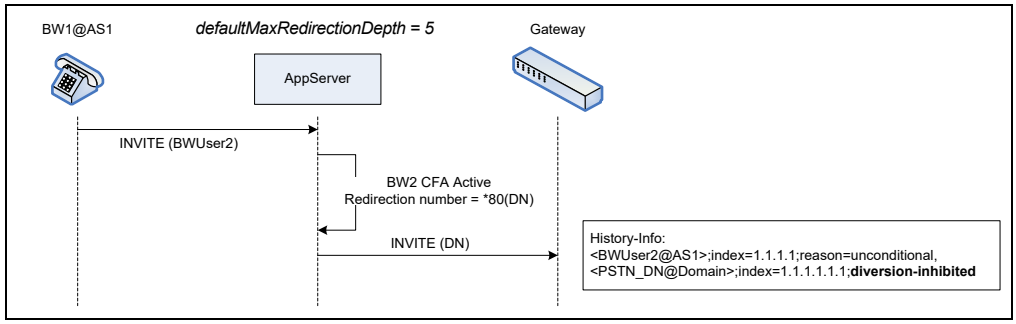

Figure 3 Diversion Inhibited with Redirection (History-Info Header)

3) If the *History-Info* header is absent or empty (this is an origination), the number of index levels is equal to the *defaultMaxRedirectionDepth + 1* and all levels are set to "1".

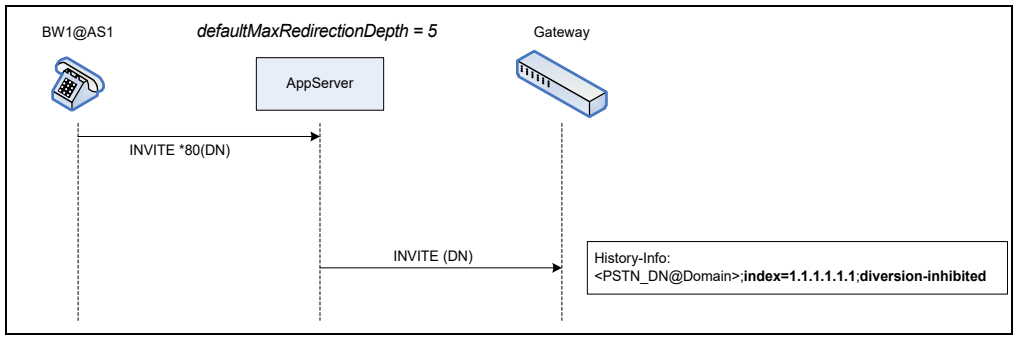

Figure 4 Diversion Inhibited on Origination (History-Info Header)

3.10.7.3 History-Info Header with Privacy Service

If a request is being forwarded to a non-trusted device or non-trusted domain, Cisco BroadWorks makes anonymous any *History-Info* entry with privacy flag (by setting the name to "Anonymous" and the SIP URI to "sip:anonymous@anonymous.invalid"). In addition, if no internal redirection occurs, Cisco BroadWorks proxies a received *Privacy history* and *History-Info* header from a trusted entity to a trusted entity. However, if a redirection occurs, upon the redirection, Cisco BroadWorks sets the individual history-info items to "private" and adds the new history-info item using the privacy restrictions of the redirecting party.

Figure 5 shows a message flow when Cisco BroadWorks receives an INVITE request with the *Privacy* header set to "history". The Application Server shows Cisco BroadWorks makes anonymous the *History-Info* header if the Privacy SIP header is set to "history", "session", or "header" prior to forwarding the INVITE request to a non-trusted device.

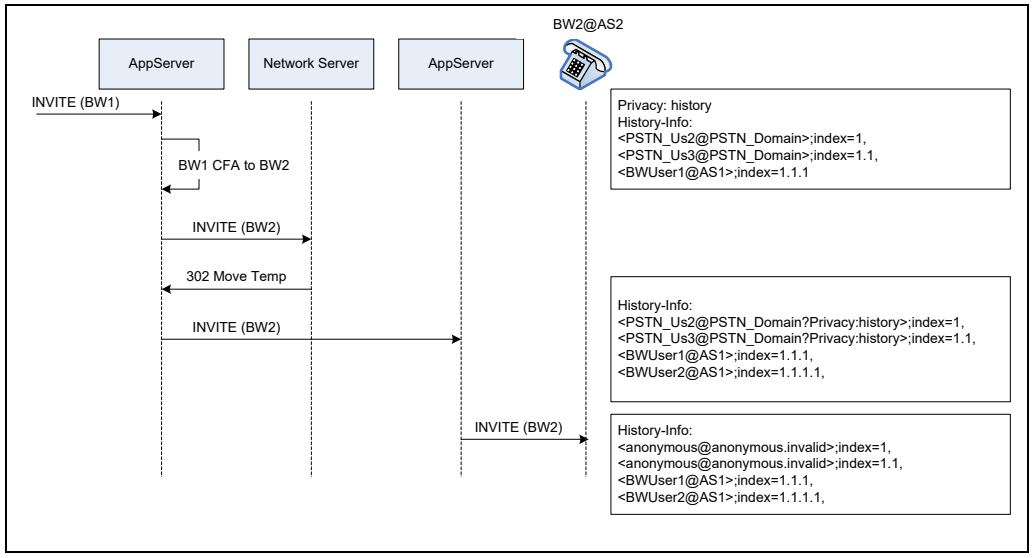

Figure 5 History-Info and Privacy Header

Figure 6 shows a message flow when Cisco BroadWorks receives an INVITE request with the *Privacy* attribute set to "history" on some entries. The Application Server shows that Cisco BroadWorks makes anonymous the History-Info entry with the *Privacy* attribute prior to forwarding the INVITE request to a non-trusted device.

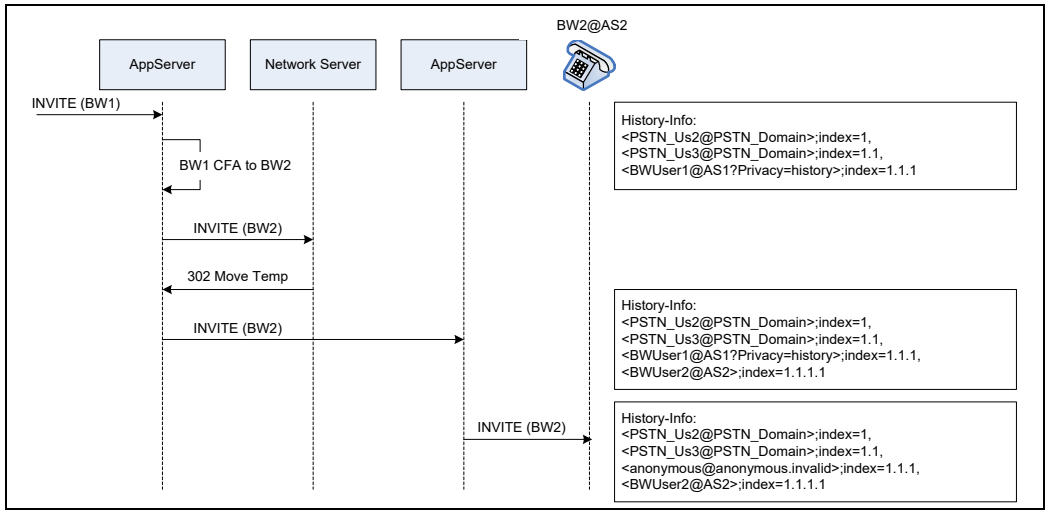

Figure 6 History-Info and Privacy Attributes

3.10.7.4 Diversion Header to History-Info Header

Figure 7 shows a message flow for a scenario in which Cisco BroadWorks converts the *Diversion* counter to the *History-Info* index. As shown, when Cisco BroadWorks receives a *Diversion* entry with a counter different from 1, a gap is created in the index.

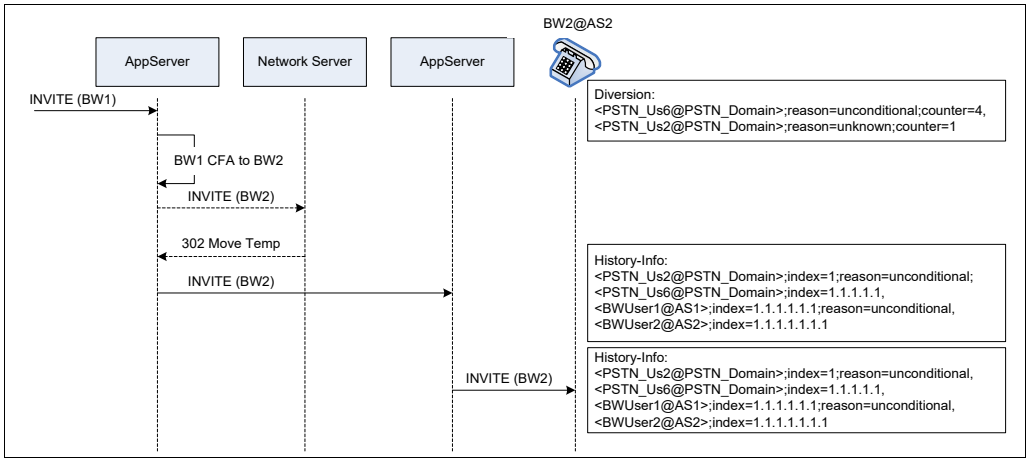

Figure 7 Counter to Index Conversion

3.10.7.5 History-Info Header to Diversion Header

Figure 8 shows a message flow for a scenario in which Cisco BroadWorks converts the *History-Info* index to the *Diversion* counter. As shown, when Cisco BroadWorks receives a *History-Info* entry with a gap in the index, the counter is more than 1.

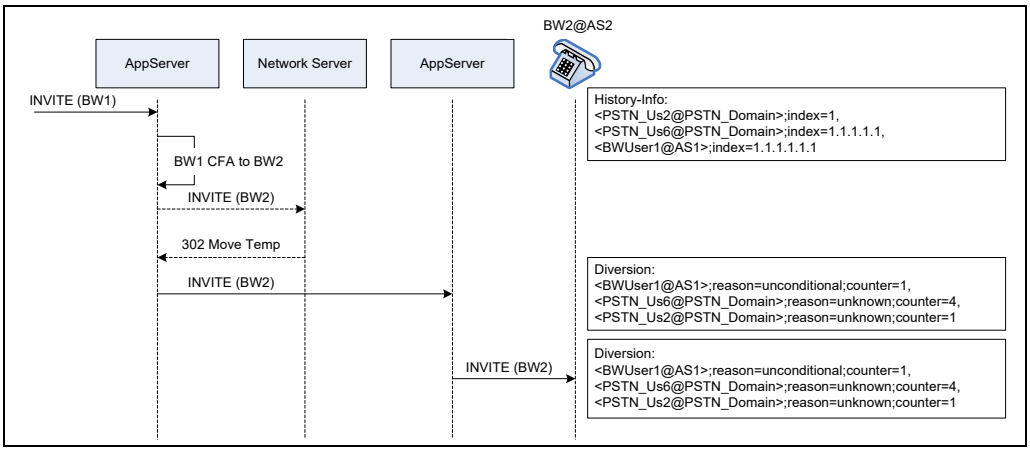

Figure 8 Index to Counter Conversion

3.10.7.6 History-Info with Cause Parameter

The following call flow illustrates diverting scenarios when UserA calls SipUser1.

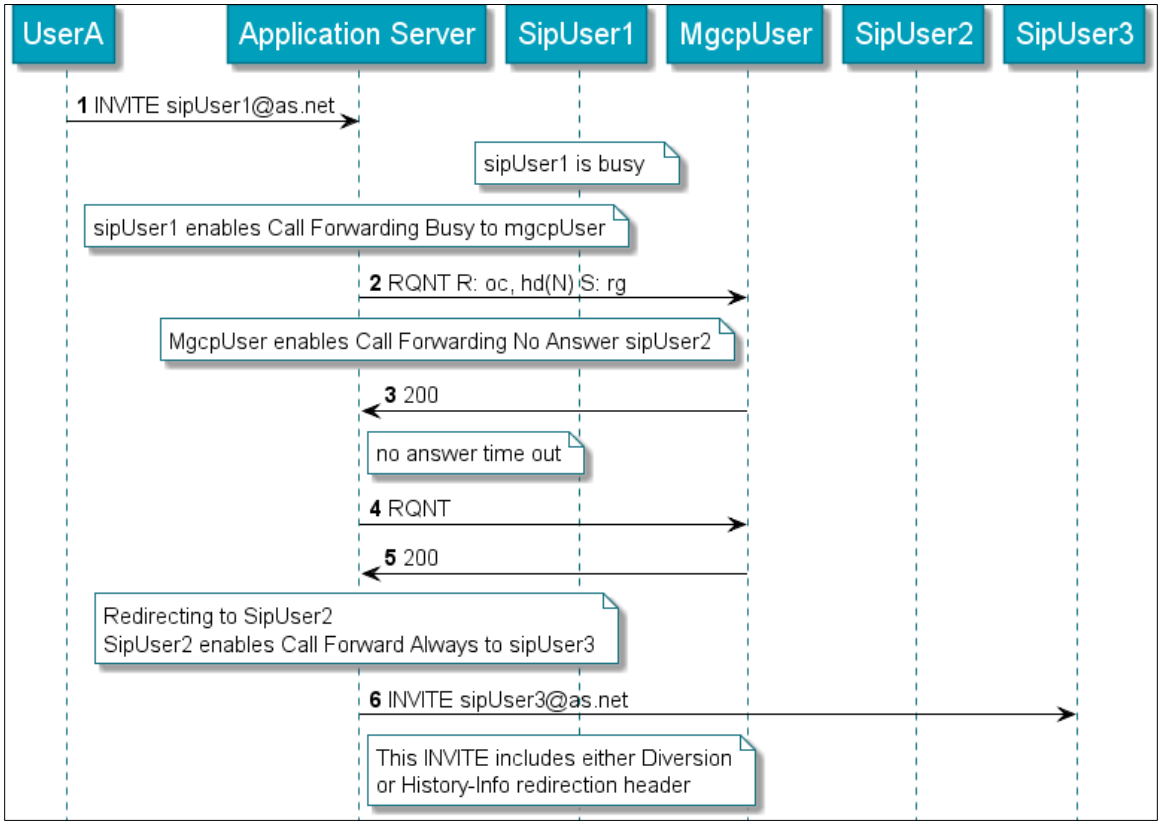

Figure 9 Cisco BroadWorks User Diversion Call Flow

When SipUser3 supports *History-Info*, message 6 has the following header content when the cause parameter is supported.

```
INVITE sip:sipUser3@as.net;user=phone;cause=302 SIP/2.0
History-Info:
<sip:sipUser1@as.net;user=phone?Reason=SIP%3btext%3d%22Busy%20Here%22%3bc
ause%3d486%2cDiversion%3btext%3d%22user-busy%22>;index=1,
<sip:mgcpUser@as.net;user=phone;cause=486?Reason=SIP%3btext%3d%22Request%
20Timeout%22%3bcause%3d408%2cDiversion%3btext%3d%22no-
answer%22>;index=1.1,
<sip:sipUser2@as.net;user=phone;cause=408?Reason=SIP%3btext%3d%22Moved%20
Temporarily%22%3bcause%3d302%2cDiversion%3btext%3d%22unconditional%22>;in
dex=1.1.1,
<sip:sipUser3@as.net:5060;user=phone;cause=302>;index=1.1.1.1
```

The previous message shows the *cause* parameter added to the *Request-URI* and to all the *History-Info* entries except for the first one.

The first entry does not have a cause parameter but it does have an embedded *Reason* header indicating that sipUser1 diverted the call due to *call forward user busy*.

```
<sip:sipUser1@as.net;user=phone?Reason=SIP%3btext%3d%22Busy%20Here%22%3bc
ause%3d486%2cDiversion%3btext%3d%22user-busy%22>;index=1,
```

The second entry has a cause=486 indicating that sipUser1 diverted the call to mgcpUser because of *user busy*. Additionally, with the embedded *Reason* header, this entry also indicates that mgcpUser diverted the call due to *call forward no answer*.

```
<sip:mgcpUser@as.net;user=phone;cause=486?Reason=SIP%3btext%3d%22Request%
20Timeout%22%3bcause%3d408%2cDiversion%3btext%3d%22no-
answer%22>;index=1.1,
```

The third entry has a cause=408 indicating that mgcpUser diverted the call to sipUser2 due to *call forward no answer*. It also indicates that sipUser2 diverted the call due to *call forward always*.

```
<sip:sipUser2@as.net;user=phone;cause=408?Reason=SIP%3btext%3d%22Moved%20
Temporarily%22%3bcause%3d302%2cDiversion%3btext%3d%22unconditional%22>;in
dex=1.1.1,
```

The fourth entry has the same URI as the Request-URI. It has a cause=302 indicating that sipUser2 diverted the call to sipUser3 due to *call forward always*.

```
<sip:sipUser3@as.net:5060;user=phone;cause=302>;index=1.1.1.1
```

When SipUser3 supports Diversion, message 6 has the following header content.

```
INVITE sip:sipUser3@as.net;user=phone SIP/2.0
Diversion:
<sip:sipUser2@as.net;user=phone;user=phone>;privacy=off;reason=unconditio
nal;counter=1,
<sip:mgcpUser@as.net;user=phone>;privacy=off;reason=no-answer;counter=1,
<sip:sipUser1@as.net;user=phone>;privacy=off;reason=user-busy;counter=1
```

3.10.7.7 History-Info in Response Message

When *includeHistoryInfoInResponse* is set to “true”, the Cisco BroadWorks Application Server accepts the *History-Info* header in the *200 OK* response to the initial INVITE from trusted endpoints.

When accepted, the *History-Info* header is proxied to the originating side with no modification and it is included in a *200 OK* sent to the originating endpoint.

When *includeHistoryInfoInResponse* is set to “true” and the received *200 OK* response does not contain a *History-Info* header or the *History-Info* header is not accepted (from an untrusted endpoint), then the *History-Info* header received in the originating INVITE, if any, is included in the *200 OK* response sent from the Cisco BroadWorks Application Server.

The following call flow illustrates a situation in which the *History-Info* header is relayed in the *200 OK* back to the originator.

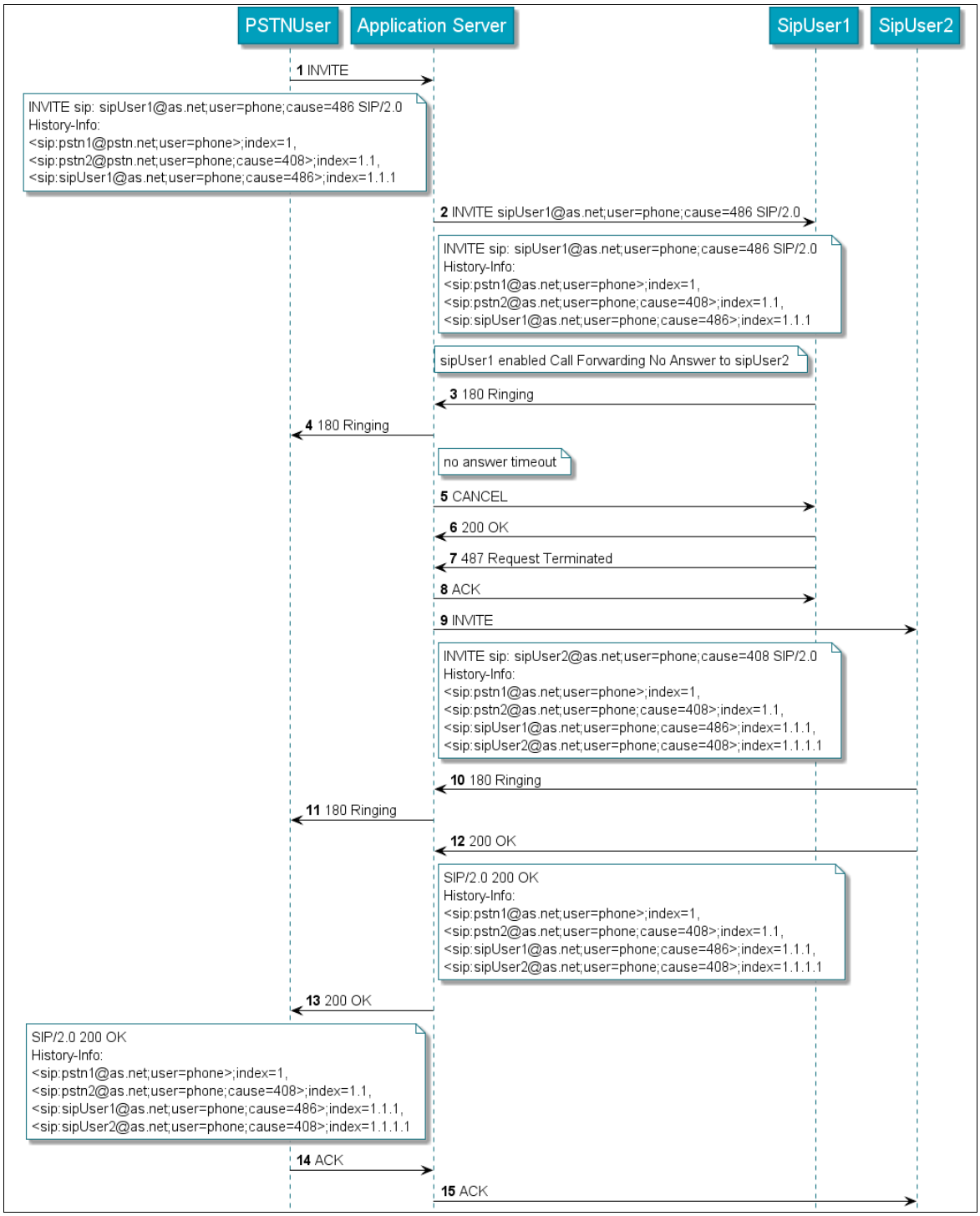

Figure 10 History-Info in 200 OK Accepted

The following call flow shows the content of the *History-Info* header sent from Cisco BroadWorks Application Server when the *History-Info* header in the *200 OK* is **not** accepted (sent from an untrusted endpoint).

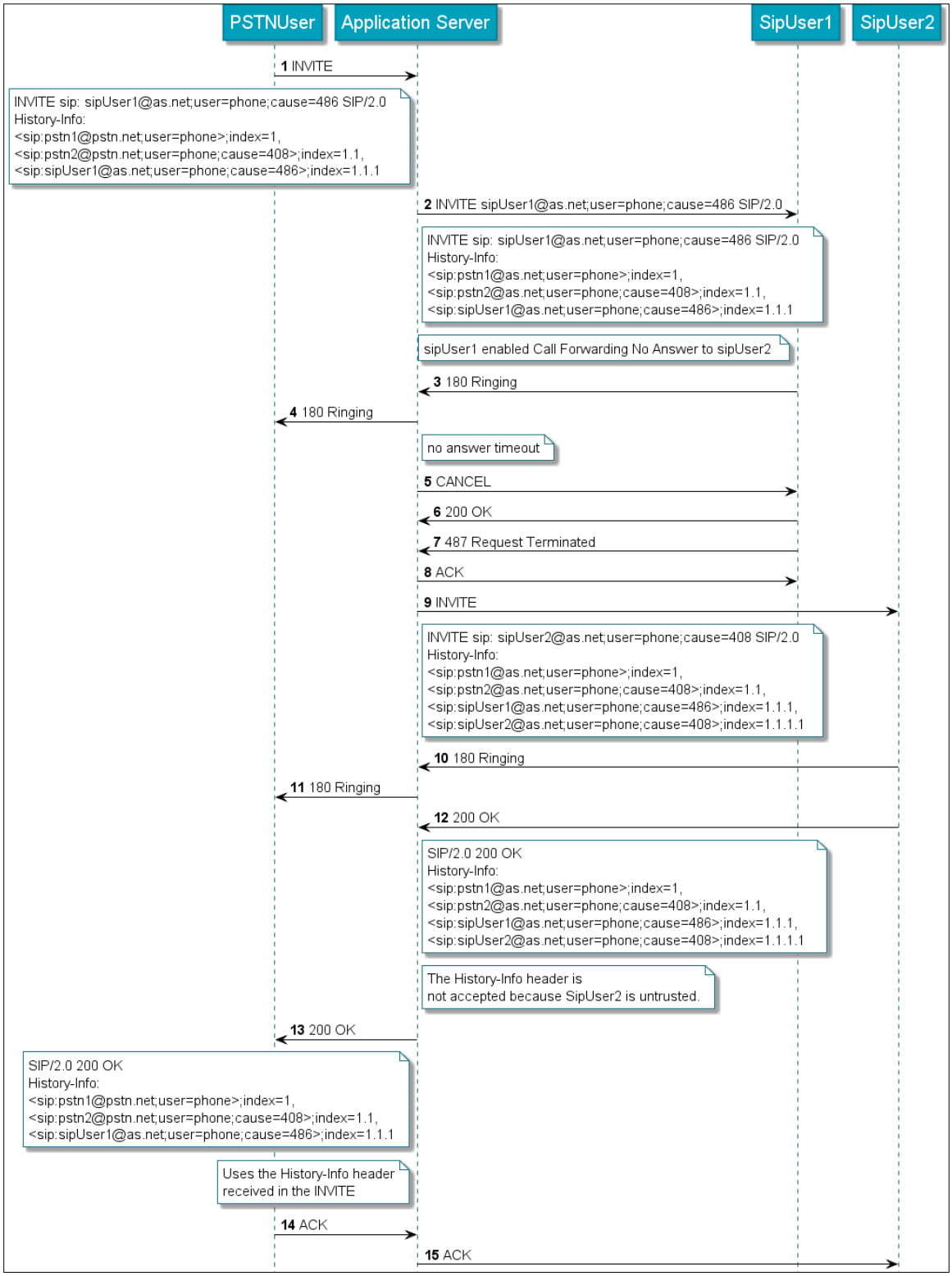

Figure 11 History-Info in 200 OK Not Accepted

3.11 Charge Header Support

Cisco BroadWorks supports the *Charge* and *P-Charge-Info* headers to facilitate interworking with GR.394-compliant ISUP networks. These headers allow Cisco BroadWorks to populate the charging information for a subscriber independently from the subscriber's calling line identity. The *Charge* or *P-Charge-Info* header used is configurable.

Note that the information populated by Cisco BroadWorks in the *Charge* or *P-Charge-Info* headers is populated by a softswitch or network gateway in the *Charge Number* parameter in the ISUP IAM according to GR.394.

When the Charge Number service is assigned, the configured charge number for the user is included in the call detail records (CDRs) generated for the user's originating calls and included in the *Charge* or *P-Charge-Info* headers of the SIP INVITEs for calls originated by the user.

The charge number is only used for calls going out to the network that are originated by a Cisco BroadWorks subscriber with the Charge Number service assigned and a charge number configured for the subscriber. When the Charge Number service is enabled, all calls include an extra field in the originating CDR containing the configured charge number for the subscriber.

Calls that are redirected to the network contain only the charge number of the last redirecting party, if it exists, but do not contain the charge number of the redirected party. For example, party A with the Charge Number service assigned calls party B with the Charge Number, and party B forwards the call to the PSTN through a softswitch or network gateway. The resulting SIP INVITE to the softswitch or network gateway includes a single *Charge* header populated with the charge number of party B.

3.11.1 Charge Header

A SIP header, *Charge,* is introduced for originating INVITEs. It contains an addr-spec and defines a single parameter, *noa* (nature of address). The syntax of the *Charge* header is as follows:

```
charge = "charge" HCOLON "<" addr-spec ">" ";" noa

noa = "noa" EQUAL "clgp-ani-natl-num"
```

For outgoing calls from a Cisco BroadWorks subscriber with the Charge Number service assigned, the *Charge* header is populated with the configured charge number of the subscriber and always includes the *noa* parameter with a value of "clgp-ani-subscriber-num". The *noa* parameter indicates to the PSTN that the charge number is set to the calling party national number.

```
INVITE sip:6137222000@47.174.73.240:5060;SIP/2.0
From: <sip:6137222010@47.174.73.240:5060>;tag=f-13c4-419b7608-137445-
144454f6
To: <sip:6137222000@47.174.73.240:5060>
Call-ID: 41840594f049ae2f13c4419b760813744521850ff89430478-0125-7168
CSeq: 1 INVITE
User-agent: CS2000/NGSS/9.0
X-Nortel-Profile: DEFAULT
Max-Forwards: 70
Supported: 100rel
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,INFO,SUBSCRIBE,REFER,NOTIFY, PRACK
Via: SIP/2.0/UDP SP2KDON:5060;maddr=47.174.73.240;branch=z9hG4bK-
419b7608-137445
P-Asserted-Identity: <sip:6137222010@47.142.211.17;user=phone>
Contact: <sip:47.174.73.240:5060>
```

```
Charge:<sip: +16137222011>;noa=clgp-ani-natl-num
```

3.11.2 P-Charge-Info Header

The *P-Charge-Info* header is defined in IETF's *draft-york-sipping-p-charge-info-06* draft document [50].

Cisco BroadWorks may send the *P-Charge-Info* header as either a sip URI or a tel URI, depending on the setting of the SIP interface parameter *chargeHeaderFormat*.

The *P-Charge-Info* header has the following syntax.

```
P-Charge-Info = "P-Charge-Info" HCOLON (name-addr / addr-spec)*
              (SEMI charge-param)
              ; name-addr and addr-spec are specified in RFC 3261
          charge-param = npi-param / noa-param / generic-param
          npi-param = "npi" EQUAL npi-value
              ; generic-param is specified in RFC 3261
          npi-value = ("ISDN" / "DATA" / "TELEX" / "PRIVATE" /
                 "SPARE0" / "SPARE1" / "SPARE2" / "SPARE3" /
                 "SPARE4" / "SPARE5" / "SPARE6" / "SPARE7" /
                 "UNKNOWN" )
          noa-param = "noa" EQUAL noa-value
          noa-value = gen-value
```

Example message:

```
INVITE sip:+19726980601@172.30.90.4:5060;user=phone SIP/2.0
Via:SIP/2.0/UDP 172.30.90.4;branch=z9hG4bKBroadWorks.1ssl2nc-
172.30.90.4V5060-0-363685998-233544608-1239825436890-
From:"john2 north"<sip:9726980502@172.30.90.4;user=phone>;tag=233544608-
1239825436890-
To:<sip:+19726980601@172.30.90.4:5060;user=phone>
Call-ID:BW145716890150409102794624 3@172.30.90.4
CSeq:363685998 INVITE
Contact:<sip:172.30.90.4:5060>
P-Asserted-Identity:"john2 north"<sip:9726980502@172.30.90.4;user=phone>
Privacy:none
Diversion:"john2
south"<sip:888999@172.30.90.4;user=phone>;privacy=off;user-
id="south02@txasdev80.net";delay-ccm;reason=follow-me;counter=1
Supported:100rel
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Accept:multipart/mixed,application/media_control+xml,application/sdp
P-Charge-Info:<tel:3331111205>;noa=clgp-ani-natl-num
Max-Forwards:10
Content-Type:application/sdp
Content-Length:112
```

3.12 E911 Support/NENA i2 Compliance

Cisco BroadWorks implements a subset of the recommendations for the V5 interface defined in the National Emergency Number Association (NENA) 911 i2 draft document. The V5 interface defines the protocol between the call server (Cisco BroadWorks) and the 911 redirect server.

Cisco BroadWorks implements the portion of the V5 interface that specifies the format of the *Request-URI* and the *P-Asserted-Identity* headers in the redirected SIP INVITE. The above-mentioned headers follow the format indicated in section *5.5.5.3* of the NENA document [41]. The NENA specification requires that the redirected SIP INVITE to the Emergency Services Gateway (ESGW) contain the emergency services routing number (ESRN) in the Request-URI and the emergency services query key (ESQK) in the *P-Asserted-Identity* header. The INVITE to ESGW obtains the Request-URI from the *Contact* header of the SIP Redirect response, and the *P-Asserted-Identity* header is built from the *P-Asserted-Identity* header contained in the *Contact* header of the SIP Redirect response (3XX Response).

```
Generated SIP INVITE:

INVITE sip:Emergency-Services-Routing-Number;user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.1.142;branch=z9
From:"aaa aaa"<sip:+12403641000@192.168.1.142;user=phone>;tag=7
To:<sip:911@RedirectServer;user=phone>
Call-ID:BW11482204223110510933850850192.168.1.142
CSeq:520309774 INVITE
Contact:<sip:192.168.1.142:5060>
P-Asserted-Identity:Emergency-Services-Query-Key>
```

Figure 12 shows the Cisco BroadWorks to Routing gateway – V5 interface.

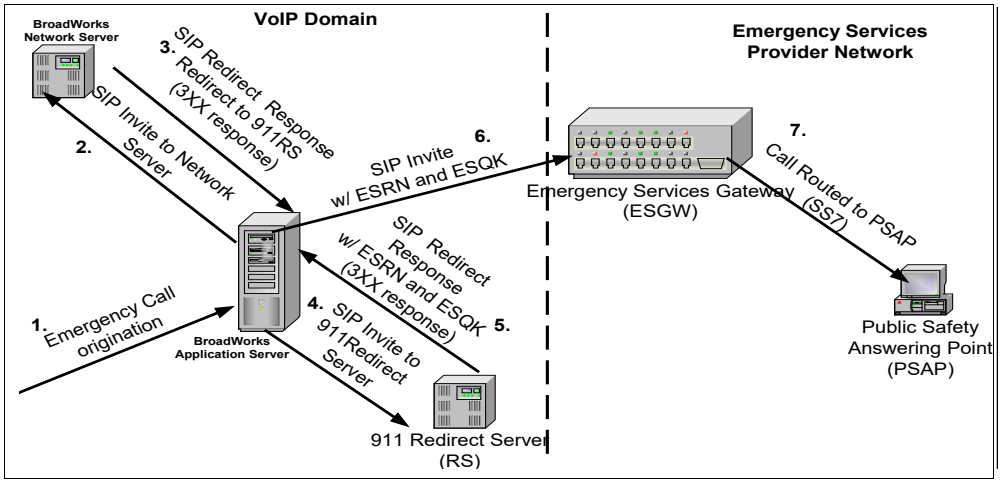

Figure 12 Cisco BroadWorks to Routing Gateway – V5 Interface

Figure 13 shows the V5 interface header mapping.

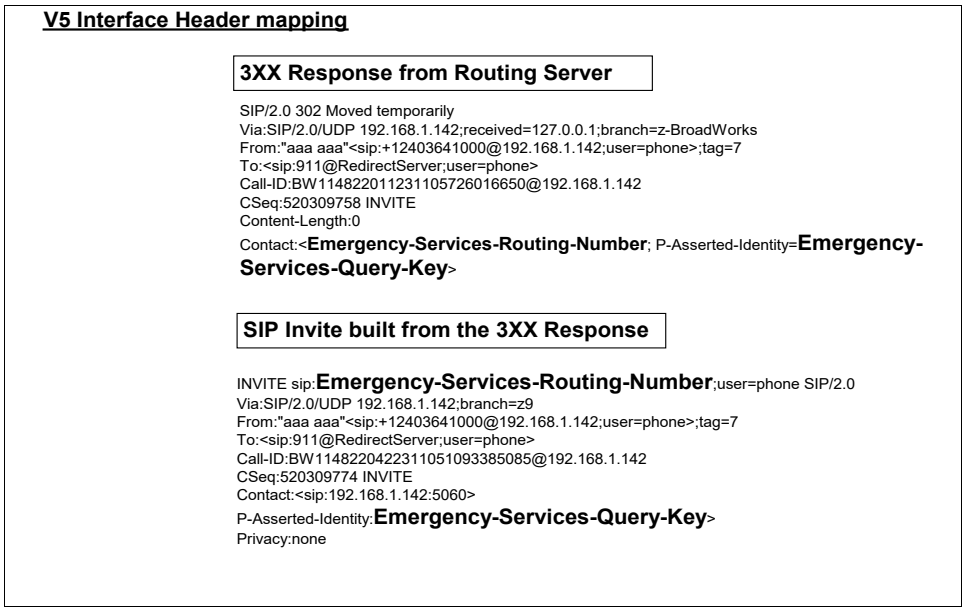

Figure 13 V5 Interface Header Mapping

Figure 14 shows the call flow with headers.

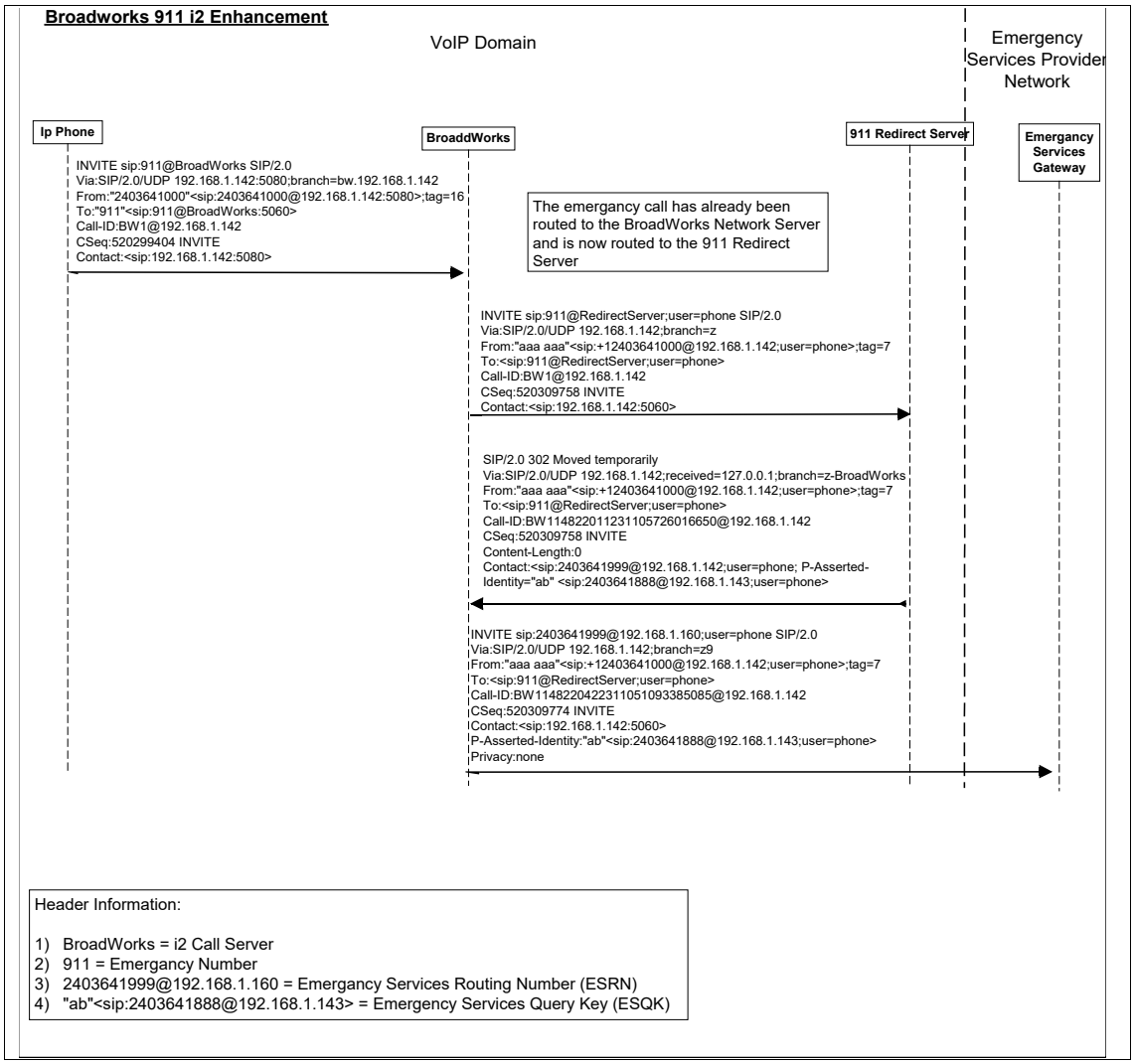

Figure 14 Call Flow with Headers

3.13 Offer/Answer Model

Reference Documents:

- *RFC 3264: An Offer/Answer Model with the Session Description Protocol, June 2002*
- *RFC 6337: Session Initiation Protocol (SIP) Usage of the Offer/Answer Model*, August 2011

3.13.1 Overview

Cisco BroadWorks fully supports the offer/answer model for exchanging SDP session descriptions, as described in *RFC 3264*. Cisco BroadWorks supports multiple early dialogs, and it tracks the offer/answer status of each individual early dialog. For devices that do not fully support *RFC 3264*, Cisco BroadWorks provides some configurable options that can enable those devices to interwork with other devices that do support *RFC 3264*.

NOTE: For Cisco BroadWorks to behave in a way that is fully compliant with *RFC 3264* the SIP system parameter *useStrictRFC3264Compliance* must be set to "true". The default value for this parameter is "false", which causes Cisco BroadWorks to operate with relaxed compliance for improved interoperability with many devices found in actual deployments.

While the offer/answer model is fundamental, it interacts with many advanced SIP options. Further discussion of the offer/answer model is provided in sections that discuss those SIP options.

- Section *3.14 SIP Forking* provides details concerning the offer/answer model as it applies to SIP forking.
- Section *3.16 Reliability of Provisional Responses in SIP (RFC 3262)* provides details concerning the offer/answer model as it applies to reliable provisional responses.
- Section *3.17 Session Initiation Protocol UPDATE Method (RFC 3311)* provides details concerning the offer/answer model as it applies to the SIP UPDATE method.

Although Cisco BroadWorks operates as a back-to-back user agent (B2BUA), it does not provide media relay. Instead, Cisco BroadWorks facilitates media negotiation between the two connected endpoints. Consequently, Cisco BroadWorks supports offer/answer end to end, rather than leg by leg.

In general, Cisco BroadWorks relays the SDP between the endpoints in a way that does not affect media negotiation. However, Cisco BroadWorks does not relay the SDP unchanged, but makes changes to the "o" line and other lines as necessary, while generally keeping the media stream negotiation transparent to the endpoint devices. Cisco BroadWorks also provides various services that can enforce certain codec policies.

3.13.2 Call Flows

RFC 6337 provides guidance on the correct usage of the offer/answer model in several SIP call scenarios. The RFC provides six different patterns for correct offer/answer exchanges. Cisco BroadWorks operates correctly for all of these six different patterns.

The following diagram provides a call flow for the first pattern presented in *RFC 6337*. The diagram shows that "[offer A]" becomes "[offer A*]", as a reminder that Cisco BroadWorks "rebrands" the SDP, while keeping the media stream negotiation largely unchanged. Likewise, "[answer A*]" becomes "[answer A]".

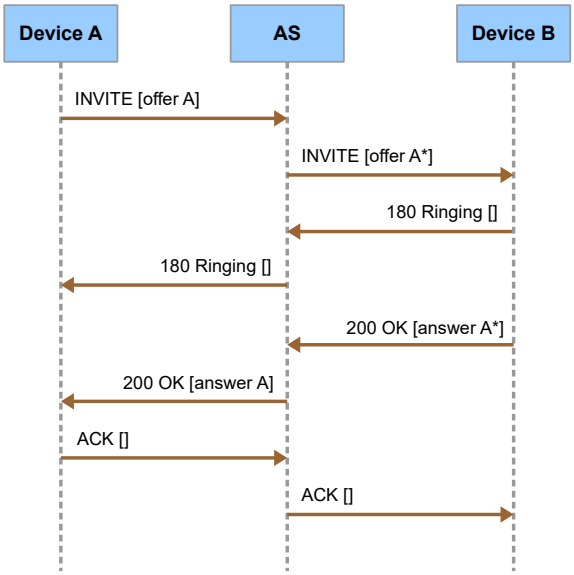

Figure 15 Basic Offer/Answer Scenario with Offer in INVITE Request

The following diagram provides a call flow for the second pattern presented in *RFC 6337*, in which the terminating endpoint sends the offer in its *200* response and the originating endpoint sends the answer in its ACK request.

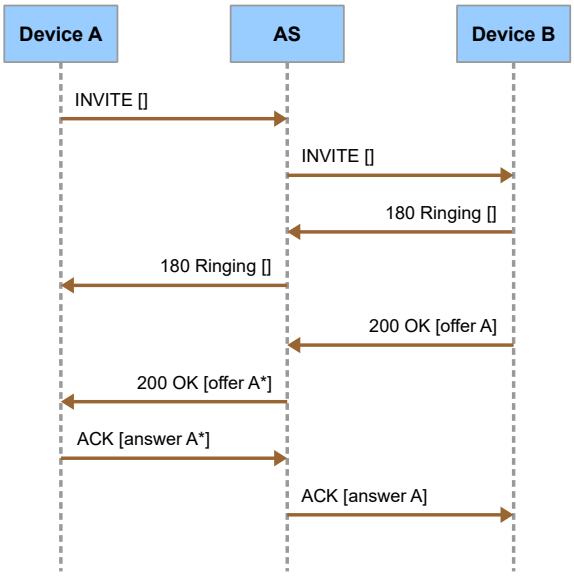

Figure 16 Basic Offer/Answer with Offer in 200 Response

In the special case where the terminating endpoint (Device B in the call flow diagram) cannot receive an initial INVITE request without SDP, Cisco BroadWorks can optionally send an initial INVITE request with a “fake” SDP. This option is enabled on the network interface when the SIP parameter *networkSupportInviteWithoutSdp* is set to “false”. The fake SDP is syntactically correct, but the provided RTP transport addresses are not addresses of true RTP endpoints. In this special scenario, Cisco BroadWorks sends a re-INVITE to the terminating device immediately after answer. The call flow is provided in the following diagram. Offer A is the fake SDP.

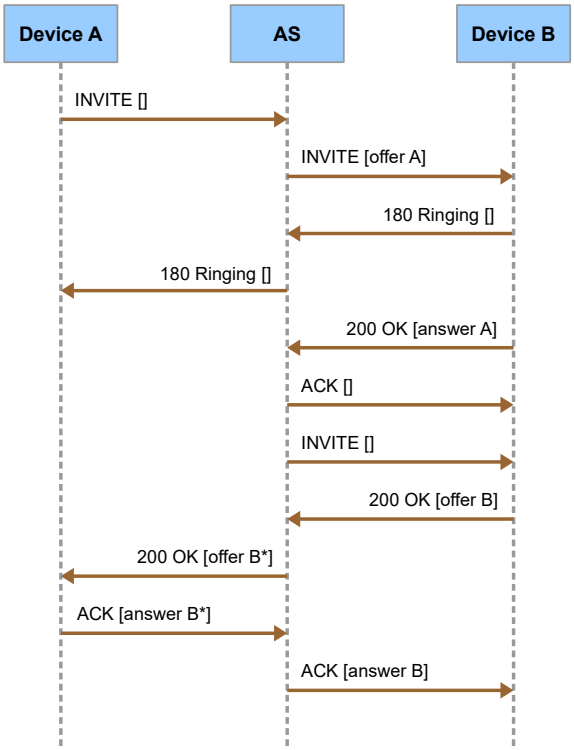

Figure 17 Basic Offer/Answer with Offer in 200 Response, Alternate Scenario

Cisco BroadWorks supports the remaining four offer/answer patterns presented in *RFC 6337*, including the patterns that involve PRACK (see section *3.16 Reliability of Provisional Responses in SIP (RFC 3262)*) and UPDATE (see section *3.17 Session Initiation Protocol UPDATE Method (RFC 3311)*).

3.14 SIP Forking

Reference Documents:

- RFC 3261: *SIP: Session Initiation Protocol*, June 2002
- RFC3841: *Caller Preferences for the Session Initiation Protocol (SIP)*
- RFC 6228: *Session Initiation Protocol (SIP) Response Code for Indication of Terminated Dialog*, May 2011

3.14.1 Overview

A User Agent Client (UAC) that conforms to *RFC 3261* must support receiving provisional responses that create multiple early dialogs. Such multiple early dialogs typically result when a downstream proxy server forks the initial INVITE request to multiple terminating endpoints. Cisco BroadWorks fully supports multiple early dialogs.

Cisco BroadWorks can interwork with network elements that support SIP forking as well as those that do not. As a Back-to-Back User Agent (B2BUA), Cisco BroadWorks can even facilitate interworking between a downstream forking proxy server and an upstream SIP network element that does not support SIP forking. Acting as a UAC, Cisco BroadWorks fully supports SIP forking by downstream proxy servers. This behavior is fully compliant to *RFC 3261* and is not configurable. Acting as a User Agent Server (UAS), Cisco BroadWorks supports several configuration options that control how it interworks with an upstream UAC, including a UAC that is non-compliant to *RFC 3261* and supports only a single early dialog. Thus, even when there is forking downstream, Cisco BroadWorks can present a single early dialog to the originating endpoint.

3.14.2 User Agent Client Behavior

When acting as a UAC, Cisco BroadWorks fully supports SIP forking. That is, when Cisco BroadWorks sends an initial INVITE request, it is able to manage multiple early dialogs created by the provisional responses or a new confirmed dialog created by the final *200 OK* response. This functionality does not depend on any configuration options.

3.14.3 User Agent Server Behavior

When acting as a UAS, Cisco BroadWorks may need to send multiple provisional responses that create multiple early dialogs. This need arises in two situations. One situation is when a downstream proxy server sends multiple provisional responses that create early dialogs. The second situation is when Cisco BroadWorks executes complex services. For example, Cisco BroadWorks might direct a new call attempt to a terminating endpoint, then execute Call Forwarding No Answer and redirect the call to a different terminating endpoint.

Ideally, all originating endpoints would conform to *RFC 3261* and support multiple early dialogs. However, non-conforming, single-dialog-only endpoints are seen in many deployments. For interoperability, Cisco BroadWorks offers configuration options to support these single-dialog-only originating endpoints. Note, though, that because Cisco BroadWorks does not provide media relay (it negotiates session parameters end to end), there are limitations – namely, in some cases Cisco BroadWorks' interactions with single-dialog-only endpoints cannot fully conform to *RFC 3264* offer/answer requirements.

When Cisco BroadWorks acts as a UAS for an initial INVITE request, it may operate in multiple-dialog mode or single-dialog mode. By default, Cisco BroadWorks operates in multiple-dialog mode. In this mode, Cisco BroadWorks may send provisional responses with different *To* tags, creating multiple early dialogs. It may also send a *200 (OK)* response with a different *To* tag, creating a new confirmed dialog. Alternatively, Cisco BroadWorks may operate in single-dialog mode. In single-dialog mode, Cisco BroadWorks may send multiple provisional responses; however, all provisional responses and the final *200 OK* response have the same *To* tag, thus creating only a single dialog.

The selection of multiple-dialog mode or single-dialog mode is controlled statically by SIP system parameters and dynamically by the "no-fork" indicator in the incoming INVITE request. Separate SIP system parameters control the access interface and the network interface, so that, for example, the access interface can operate in multiple-dialog mode and the network interface in single dialog mode.

Whether operating in multiple-dialog mode or single-dialog mode, Cisco BroadWorks supports "sub-modes" that further refine Cisco BroadWorks behavior. In multiple-dialog mode, Cisco BroadWorks supports these sub-modes:

- Multiple Dialogs – In this sub-mode, Cisco BroadWorks assumes the UAC supports multiple dialogs in conformance with *RFC 3261*. Cisco BroadWorks does not perform the error correction described below.
- Multiple Dialogs With Error Correction – In this sub-mode, Cisco BroadWorks assumes the UAC supports multiple early dialogs in conformance with *RFC 3261*. Cisco BroadWorks may perform the error correction described below. This is the default sub-mode when Cisco BroadWorks operates in multiple-dialog mode.

The difference between the two multiple-dialog sub-modes concerns the way Cisco BroadWorks handles a particular scenario in which the terminating endpoint violates the offer/answer SDP negotiation. As *RFC 6337* explains, the terminating endpoint may send the answer SDP in a provisional response, followed by the same SDP in the final *200 OK* response. However, if the terminating endpoint sends one SDP in a provisional response, followed by a different SDP in the *200 OK* response within the same dialog, then the endpoint violates the offer/answer protocol by sending two different answers for one offer. Cisco BroadWorks can correct this violation or expose it to the originating endpoint. If the sub-mode is "Multiple Dialogs With Error Correction", then Cisco BroadWorks corrects the violation, sending the *200 OK* response with a different *To* tag and creating a new confirmed dialog. If the sub-mode is "Multiple Dialogs", then Cisco BroadWorks sends the *200 OK* response with the same *To* tag, exposing the originating endpoint to the offer/answer violation. A similar scenario exists where the terminating endpoint sends different SDPs in successive provisional responses. If the sub-mode is "Multiple Dialogs With Error Correction", then Cisco BroadWorks corrects this violation by sending provisional responses with different *To* tags to the originating endpoint.

In single-dialog mode, Cisco BroadWorks support these sub-modes:

- Single Dialog – In this sub-mode, Cisco BroadWorks assumes the UAC supports only a single dialog. If Cisco BroadWorks needs to send an updated SDP before answer or at answer, it may do so in a new provisional response or in the *200 OK* response. All provisional responses and the final *200 OK* response have the same *To* tag. Cisco BroadWorks will not send an updated SDP in a UPDATE request.

- Single Dialog With UPDATE – In this sub-mode, Cisco BroadWorks assumes the UAC supports only a single dialog. If Cisco BroadWorks needs to send an updated SDP before answer, it does so in an UPDATE request within the early dialog. Cisco BroadWorks sends the UPDATE request even if the UAC did not indicate that it supports the UPDATE method (by including "UPDATE" in the *Allow* header). If Cisco BroadWorks needs to send an updated SDP at answer, it sends the *200 (OK)* response with the same *To* tag.
- Single Dialog With UPDATE If Allowed – In this sub-mode, Cisco BroadWorks assumes the UAC supports only a single dialog. If Cisco BroadWorks needs to send an updated SDP before answer, it does so in an UPDATE request within the early dialog if the UAC supports the UPDATE method. (To indicate that it supports UPDATE, the UAC includes "UPDATE" in the *Allow* header.) If the UAC does not support UPDATE, then Cisco BroadWorks sends an updated SDP in a provisional response with the same *To* tag. If Cisco BroadWorks needs to send an updated SDP at answer, it sends the 200 (OK) response with the same *To* tag.

The static configuration for Cisco BroadWorks access-side forking mode is controlled by the following SIP system parameters:

Name	Description
accessForkingSupport	This parameter controls Cisco BroadWorks SIP UAS forking behavior on the access interface. If the parameter is set to "singleDialog" then Cisco BroadWorks operates in single-dialog mode on the access interface. If the parameter is set to "multipleDialogs", the Cisco BroadWorks operates in multiple-dialog mode on the access interface. The default value is "multipleDialogs".
accessSingleDialogBehavior	This parameter controls Cisco BroadWorks single dialog behavior on the access interface. The possible values are "singleDialog", "singleDialogWithUpdate", and "singleDialogWithUpdateIfAllowed". The default value is "singleDialogWithUpdateIfAllowed".
accessMultipleDialogBehavior	This parameter controls Cisco BroadWorks multiple dialog behavior on the access interface. The possible values are "multipleDialogs" and "multipleDialogsWithErrorCorrection". The default value is "multipleDialogsWithErrorCorrection".

The static configuration for Cisco BroadWorks network-side forking mode is controlled by the following SIP system parameters:

Name	Description
networkForkingSupport	This parameter controls Cisco BroadWorks SIP UAS forking behavior on the network interface. If the parameter is set to "singleDialog" then Cisco BroadWorks operates in single-dialog mode on the network interface. If the parameter is set to "multipleDialogs", the Cisco BroadWorks operates in multiple-dialog mode on the network interface. The default value is "multipleDialogs".
networkSingleDialogBehavior	This parameter controls Cisco BroadWorks single dialog behavior on the network interface. The possible values are "singleDialog", "singleDialogWithUpdate", and "singleDialogWithUpdateIfAllowed". The default value is "singleDialogWithUpdateIfAllowed".
networkMultipleDialogBehavior	This parameter controls Cisco BroadWorks multiple dialog behavior on the network interface. The possible values are "multipleDialogs" and "multipleDialogsWithErrorCorrection". The default value is "multipleDialogsWithErrorCorrection".

When Cisco BroadWorks is statically configured to operate in multiple dialog mode, it is still possible for the UAC to force Cisco BroadWorks to operate in single dialog mode. This behavior is configurable and is controlled by the SIP system parameter *supportNoForkOption*.
If *supportNoForkOption* is set to "true" and the originating endpoint sends an initial INVITE request with *no-fork* in the *Request-Disposition* header, then Cisco BroadWorks operates in single-dialog mode for that call, behaving according to the single dialog sub-mode.

> **NOTE:** The *no-fork* directive does not prevent Cisco BroadWorks from executing forking services such as Shared Call Appearance. However, it does force Cisco BroadWorks to operate in single-dialog mode for the call, so that it appears to the originating endpoint that the INVITE request was not forked.

When Cisco BroadWorks operates in single-dialog mode, it may face some difficult situations due to a mismatch between multiple dialogs in the terminating session and a single dialog in the originating session. When Cisco BroadWorks operates in multiple dialog mode, if faces fewer difficult situations since it can create a one-to-one correspondence between dialogs in the terminating session and dialogs in the originating session. (See *Figure 18)*. However, in single-dialog mode, Cisco BroadWorks must select a single dialog in the terminating session to associate with the single dialog in the originating session. (See *Figure 19*). Cisco BroadWorks designates one of the terminating session dialogs as the *current* dialog. It associates this current dialog with the dialog in the originating session. If the originating endpoint sends a SIP request, such as an UPDATE or INFO request in the early dialog, Cisco BroadWorks may forward the request to the endpoint connected to the current dialog.

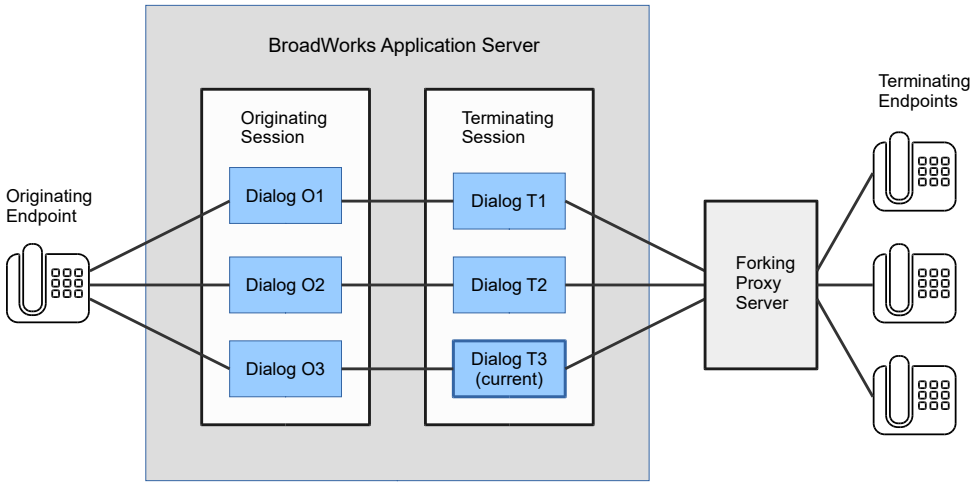

Figure 18 Originating Session with Multiple Dialog Support

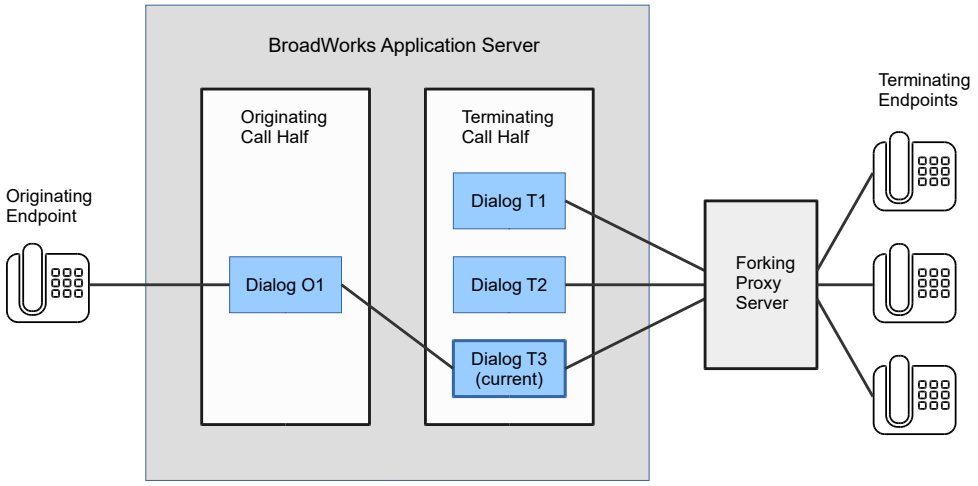

Figure 19 Originating Session with Single Dialog Support

Whenever Cisco BroadWorks creates a new dialog in the terminating session, it selects that dialog as the current dialog. Thus, the current dialog is usually the last dialog created. However, if Cisco BroadWorks receives a *199 Dialog Terminated* response for the current dialog, indicating that the dialog should terminate, it selects a different dialog to be the current dialog. The selection procedure is undocumented (and unpredictable).

When Cisco BroadWorks receives a *200 (OK)* response for one of the early dialogs, it makes that dialog a confirmed dialog and destroys all other early dialogs. Likewise, if Cisco BroadWorks receives a *200* response for a new dialog, it creates a confirmed dialog and destroys all early dialogs. If Cisco BroadWorks receives a new request, such as an UPDATE request, for a destroyed early dialog, it sends a *481 Call/Transaction Does Not Exist* response.

3.14.4 199 Provisional Response

A proxy server can send a *199 (Dialog Terminated)* provisional response to the UAC to signal that an early dialog (and an associated early media session) should be terminated. This response provides useful information to the UAC when the UAC is managing multiple early dialogs, particularly when the early dialogs involve early media. Cisco BroadWorks may be configured to support this functionality. By default, Cisco BroadWorks support for *199* responses is disabled.

Support for 199 provisional responses is enabled when the SIP system parameter *support199* is set to “true”. When 199 support is enabled, Cisco BroadWorks behavior is summarized by the following points:

- If the remote UAC indicates that it supports *199* responses (by including “199” in the *Supported* header), then Cisco BroadWorks may send *199* responses to that UAC. Note that if Cisco BroadWorks is operating in single dialog mode, it will never send a 199 response.
- Cisco BroadWorks includes “199” in the *Supported* header in the new INVITE requests it sends, indicating to the remote UAS that it supports receiving 199 responses.

- If Cisco BroadWorks receives a *199* response, it terminates the identified dialog. Cisco BroadWorks also relays the *199* response to the originating endpoint, if it is allowed. (Cisco BroadWorks relays the *199* response if the originating endpoint supports *199* responses and Cisco BroadWorks is operating in multiple dialog mode for the call.)
- Since Cisco BroadWorks operates as a B2BUA, it may receive a failure response such as *408* from a terminating endpoint, then send a 199 to the originating endpoint. This scenario may arise when Cisco BroadWorks forks an INVITE request to multiple endpoints.

Support for *199* provisional responses is disabled when the SIP system parameter *support199* is set to "false", which is the default value. When the support is disabled, Cisco BroadWorks behavior is summarized by the following points:

- Cisco BroadWorks does not send any *199* responses, even if the originating endpoint indicates it supports 199.
- If the originating endpoint sends an INVITE request with "199" in the *Require* header, then Cisco BroadWorks sends a *420 Bad Extension* response to reject the call. This response includes an *Unsupported* header with the value "199".
- Cisco BroadWorks does not send *Supported* with "199" in any outgoing INIVTE request, even if it received *Supported* with "199" in an incoming INVITE request.
- Cisco BroadWorks ignores all *199* responses.

3.14.5 Cisco BroadWorks Forking Services

Cisco BroadWorks supports several terminating user services that may fork an INVITE request to multiple endpoints. These services provide users with secondary endpoints, which are subordinate to the user's primary endpoint. These services include (among others) the following services:

- Shared Call Appearance
- BroadWorks Mobility
- BroadWorks Anywhere
- Simultaneous Ring

By default, Cisco BroadWorks relays provisional responses from a user's primary endpoint and consumes provisional responses from secondary endpoints. Thus, by default Cisco BroadWorks hides this forking activity from the originating endpoint. However, this behavior is configurable via the SIP system parameter *proxyForkingProvisionalResponses*. When *proxyForkingProvisionalResponses* is set to "false", the default value, Cisco BroadWorks consumes provisional responses from secondary endpoints. When *proxyForkingProvisionalResponses* is set to "true", Cisco BroadWorks relays provisional responses from the secondary endpoints, provided other conditions are satisfied. This behavior exposes the forking activity to the originating endpoint, which can improve the management of early media and preconditions negotiation.

Cisco BroadWorks relays provisional responses from secondary endpoints under the following conditions:

- The SIP system parameter *proxyForkingProvisionalResponses* is set to "true".
- And, the SIP system parameter *supportPEarlyMediaHeader* is set to "true".
- And, Cisco BroadWorks operates in multiple dialog mode toward the originating endpoint.

When Cisco BroadWorks hides forking activity from the originating endpoint, it modifies the SDP to secondary endpoints to prevent early media from those endpoints.

More information about Cisco BroadWorks forking services is provided in section *3.14 SIP Forking*, which describes forking in the context of early media.

3.14.6 Call Flows

The following call flow diagram shows a Cisco BroadWorks forking scenario in which Cisco BroadWorks operates in multiple-dialog mode toward the originating endpoint. In this particular scenario, Cisco BroadWorks executes Call Forwarding No Answer. Cisco BroadWorks initially sends an INVITE request to Endpoint B. When Cisco BroadWorks receives the *183 Session Progress* response from Endpoint B, it creates a new early dialog in the terminating session, then creates a new early dialog in the originating session and sends a *183* response to the originating endpoint. After the "No Answer" timer fires, Cisco BroadWorks sends a CANCEL request to Endpoint B and sends an INVITE request to Endpoint C. When Cisco BroadWorks receives the *183* response from Endpoint C, it creates a second early dialog in the terminating session, then a second early dialog in the originating session. Cisco BroadWorks relays the *183* response to the originating endpoint with a different *To* tag, so that the originating endpoint also creates a new early dialog. When Endpoint C sends the *200 (OK)* response, Cisco BroadWorks changes the dialog status from "early" to "confirmed" and relays the response to the originating endpoint.

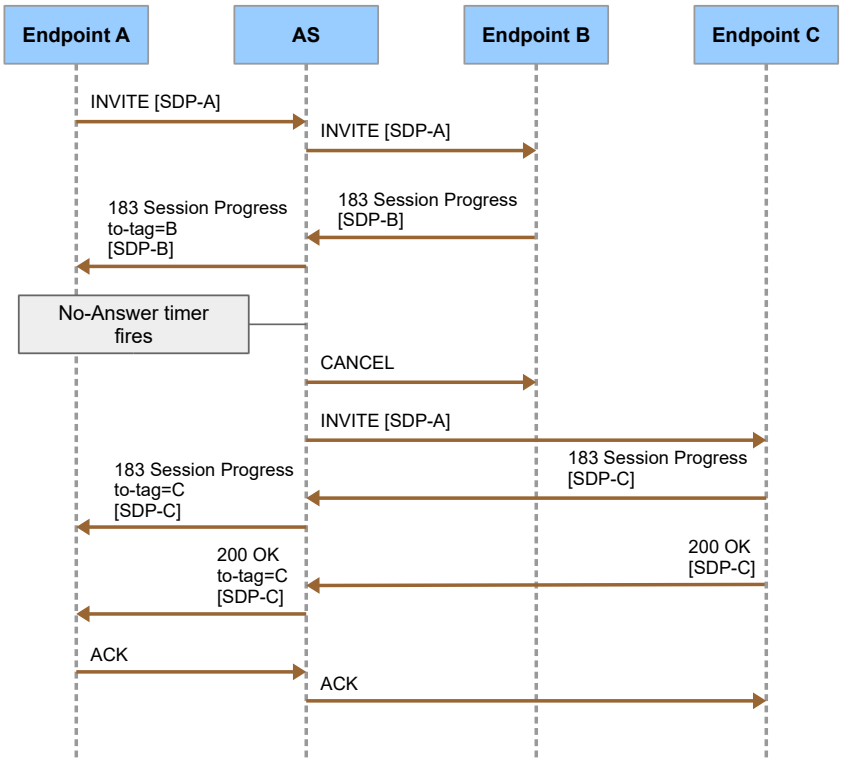

Figure 20 Call Forwarding No Answer, multiple dialogs

The following call flow diagram shows a Cisco BroadWorks forking scenario in which Cisco BroadWorks operates in single-dialog mode toward the originating endpoint. In this particular scenario, Cisco BroadWorks executes Call Forwarding No Answer. Cisco BroadWorks initially sends an INVITE request to Endpoint B. When Cisco BroadWorks receives the *183 Session Progress* response from Endpoint B, it creates a new early dialog in the terminating session, then creates a new early dialog in the originating session and sends a provisional response to the originating endpoint. After the "No Answer" timer fires, Cisco BroadWorks sends a CANCEL request to Endpoint B and sends an INVITE request to Endpoint C. When Cisco BroadWorks receives the *183* response from Endpoint C, it creates a second early dialog in the terminating session and makes this dialog the "current" dialog. Cisco BroadWorks then associates this current dialog with the early dialog in the originating session. Cisco BroadWorks relays the *183* response to the originating endpoint with the same *To* tag, so that the originating endpoint receives the response in the same early dialog. When Endpoint C sends the *200 OK* response, Cisco BroadWorks changes the dialog status from "early" to "confirmed" and relays the response to the originating endpoint.

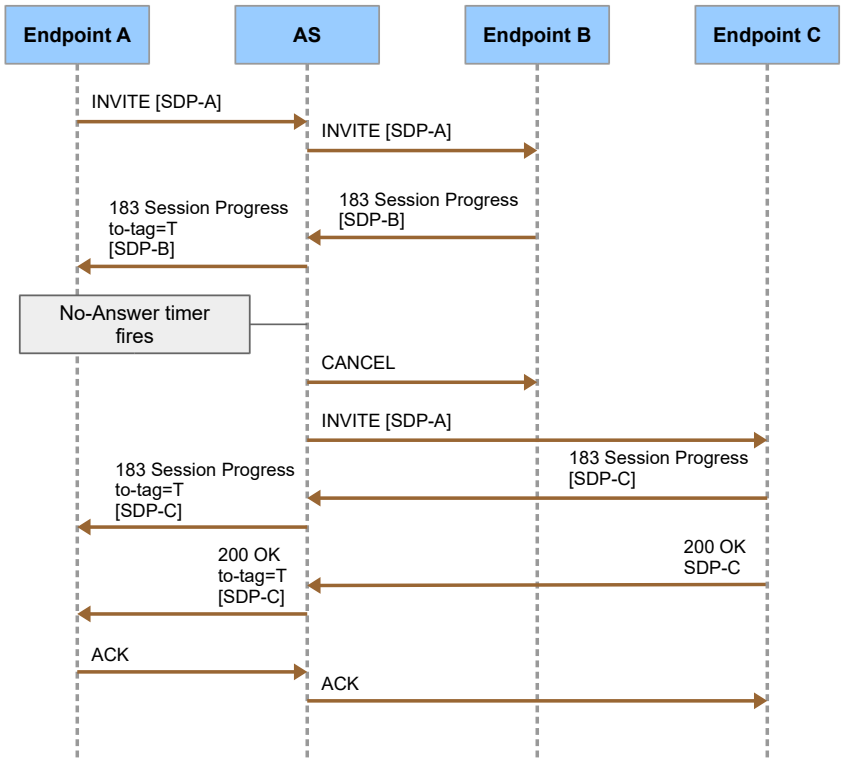

Figure 21 Call Forwarding No Answer, single dialog

The following call flow diagram shows a Cisco BroadWorks forking scenario in which Cisco BroadWorks operates in single-dialog mode and sends an UPDATE request with the new SDP. In this particular scenario, Cisco BroadWorks executes Call Forwarding No Answer. Cisco BroadWorks initially sends an INVITE request to Endpoint B. When Cisco BroadWorks receives the *183 Session Progress* response from Endpoint B, it creates a new early dialog in the terminating session, then creates a new early dialog in the originating session and sends a provisional response to the originating endpoint. After the "No Answer" timer fires, Cisco BroadWorks sends a CANCEL request to Endpoint B and sends an INVITE request to Endpoint C. When Cisco BroadWorks receives the 183 response from Endpoint C, it creates a second early dialog in the terminating session and makes this dialog the "current" dialog. Cisco BroadWorks then associates this current dialog with the early dialog in the originating session. Cisco BroadWorks sends an UPDATE request with SDP within this dialog to the originating endpoint. Endpoint A receives the new SDP as a new offer SDP and sends a *200* response with a new answer SDP. When Endpoint C sends the *200 OK* response, Cisco BroadWorks changes the dialog status from "early" to "confirmed" and relays the response to the originating endpoint. Finally, Cisco BroadWorks sends re-INVITE requests to Endpoint C and Endpoint A to perform a new offer/answer exchange.

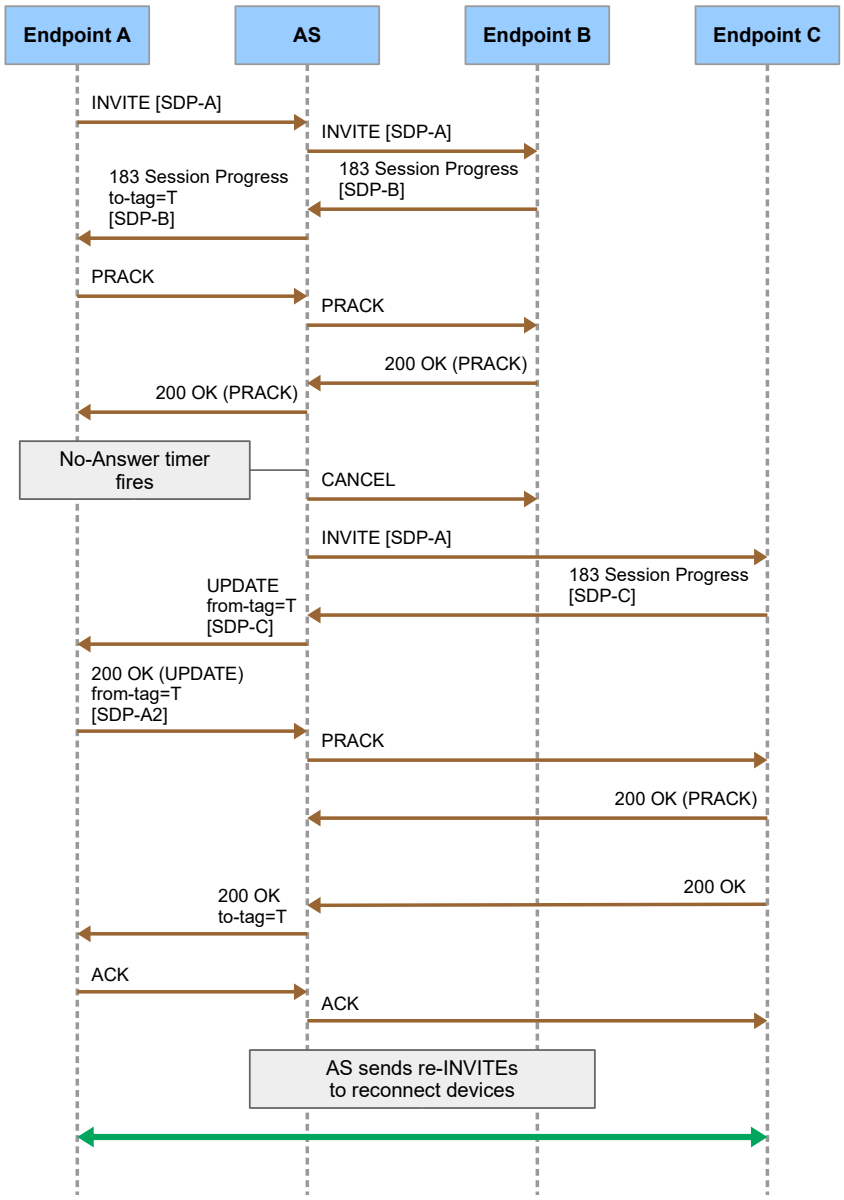

Figure 22 Call Forwarding No Answer, Single Dialog with UPDATE

The following are some additional comments about the UPDATE scenario:

- Cisco BroadWorks sends the UPDATE request with a new offer SDP. To comply with the rules of the offer/answer model, Cisco BroadWorks must not send the UPDATE request with offer SDP until any previous offer/answer exchange is completed. Therefore, Cisco BroadWorks sends the UPDATE request only if the preceding provisional response with the answer SDP is a reliable provisional response.
- The SDP from Endpoint C is an answer SDP. However, Endpoint A receives the SDP as an offer SDP in the UPDATE request. For a proper offer/answer exchange, Cisco BroadWorks sends re-INVITE requests to Endpoint C and Endpoint A immediately after answer.

3.15 Early Media Transitions

Reference Documents:

- *RFC 3398: Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping*, December 2002

3.15.1 Overview

An early media transition occurs when a terminating endpoint begins sending early media (for example, remote ringback), then stops sending early media, requiring a transition to local ringback.

When a terminating endpoint sends Cisco BroadWorks an initial provisional response with answer SDP, followed by a second provisional response in the same early dialog without SDP, there are different ways Cisco BroadWorks can interpret the second provisional response. By default, Cisco BroadWorks interprets the second provisional response to mean that the terminating endpoint ceased sending early media. In reaction to this, Cisco BroadWorks (via the Media Server) provides ringback to the originating device. This scenario is shown in the following call flow diagram.

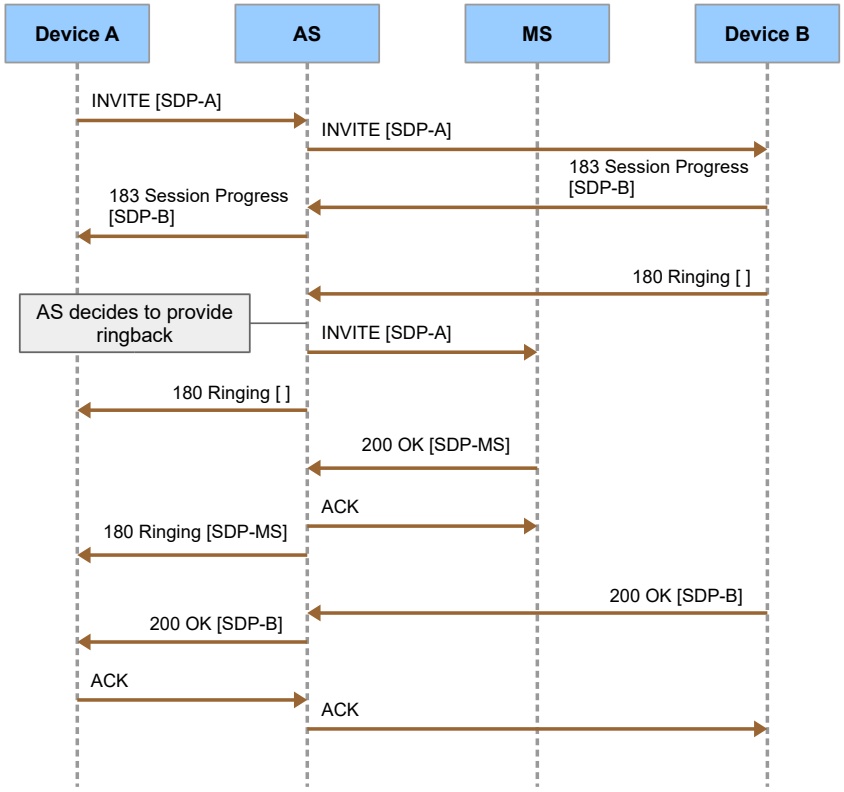

Figure 23 Early Media Transition - RFC 3398 Support Disabled

Remarks:

- Cisco BroadWorks supports reliable provisional responses. The call flow diagram shows unreliable provisional responses. However, Cisco BroadWorks' early media behavior is the same when the provisional responses are reliable.

- If Device A supports multiple dialogs, then Cisco BroadWorks sends the *180* response with SDP in a new early dialog, simulating SIP forking. If Device A does supports only a single dialog, then Cisco BroadWorks sends this response in the existing early dialog. For details on multiple dialog and single dialog support, see *3.14 SIP Forking*.

If the terminating endpoint is an ISUP gateway that follows *RFC 3398*, then it continues to send early media (for example, remote ringback) after it sends the second provisional response. In this case, Cisco BroadWorks should interpret the second provisional response as an indication of progress only and avoid providing ringback itself. This behavior is enabled when the SIP parameter *supportRFC3398* is set to "true". The scenario is shown in the following call flow diagram.

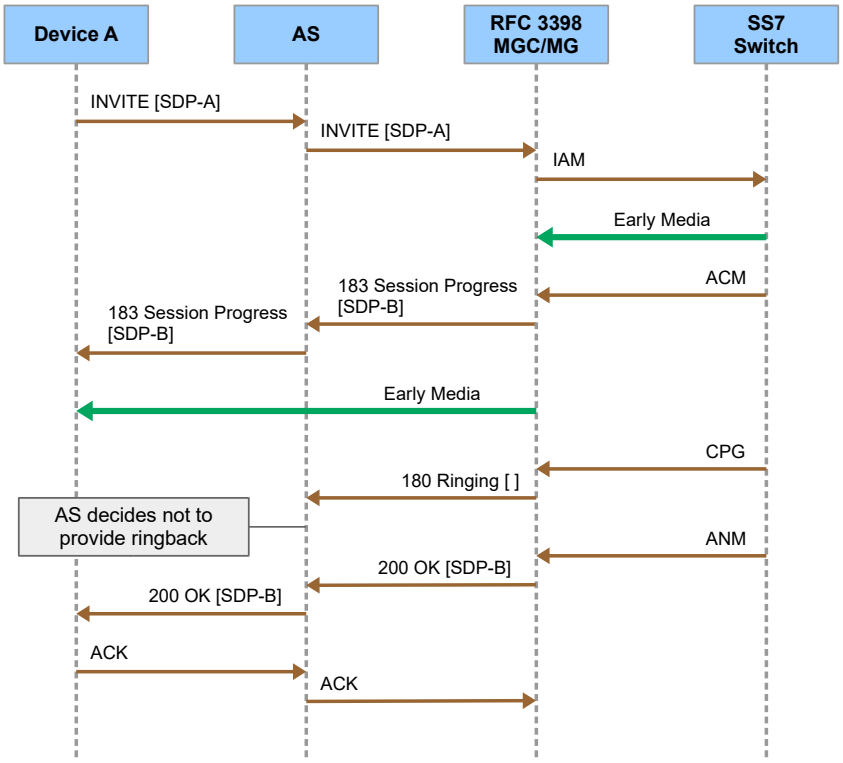

Figure 24 Early Media Transition - RFC 3398 Support Enabled

Cisco BroadWorks behavior in these scenarios is controlled by a configurable parameter, which determines whether Cisco BroadWorks supports the *RFC 3398* scenario. For the network interface, the behavior is controlled by the SIP system parameter *supportRfc3398*. For the access interface, the behavior is controlled for a specific device profile type by the parameter "Support RFC 3398" in the Identity/Device Profile Type Modify CommPilot web page. If *RFC 3398* support is disabled (the default), Cisco BroadWorks provides ringback as in the first scenario above. If *RFC 3398* support is enable, Cisco BroadWorks consumes the second provisional response as in the second scenario above.

3.15.2 Interactions with SIP Forking

In cases of SIP forking, more than one terminating endpoint may send one or more provisional responses. In this situation, Cisco BroadWorks tracks the provisional responses separately for each terminating endpoint, that is, for each early dialog. Cisco BroadWorks will provide ringback for at most one early dialog. If the UAS side operates in single dialog mode, then Cisco BroadWorks can detect an early media transition only for the current dialog.

3.15.3 Interaction with Reliable Provisional Responses

Cisco BroadWorks early media behavior as described in this section is unchanged when the terminating endpoint sends reliable provisional responses.

3.15.4 Interactions with SIP P-Early-Media Header

If Cisco BroadWorks is configured to support the *P-Early-Media* header and the provisional responses from the terminating endpoint contain a *P-Early-Media* header, then Cisco BroadWorks ignores the RFC 3398 configuration for that call. In this case, Cisco BroadWorks assumes the use of the *P-Early-Media* header alone should be sufficient to handle early media correctly. For details, see section *3.35 Cisco BroadWorks P-Early-Media Header Support (RFC 5009)*.

3.16 Reliability of Provisional Responses in SIP (RFC 3262)

Reference Documents:

- *RFC 3262: Reliability of Provisional Responses in the Session Initiation Protocol (SIP)*, June 2002
- *RFC 3264: An Offer/Answer Model with the Session Description Protocol*, June 2002
- *RFC 6337: Session Initiation Protocol (SIP) Usage of the Offer/Answer Model*, August 2011

3.16.1 Overview

Cisco BroadWorks supports reliable provisional responses as specified in *RFC 3262*. This support is enabled when the SIP system parameter *100rel* is set to “true” and disabled when it is set to “false”. By default, *100rel* is set to “true”. This section describes Cisco BroadWorks behavior when it is configured to support reliable provisional responses.

Reliable provisional responses are optional in SIP. Supporting devices negotiate the use of reliable provisional responses via the “100rel” option tag in the *Supported* or *Require* header. When Cisco BroadWorks receives *Supported* with “100rel” in an INVITE request from an originating endpoint, it also includes *Supported* with “100rel” in the INVITE request to the terminating endpoint. Conversely, if Cisco BroadWorks does not receive “100rel” in the incoming INVITE request, then it omits “100rel” in the outgoing INVITE request. In this way, Cisco BroadWorks acts similarly to a proxy server.

When Cisco BroadWorks receives a reliable provisional response from the terminating endpoint, it relays the response to the originating endpoint as a reliable provisional response. Cisco BroadWorks relays all PRACK requests, as well as the responses to PRACK requests, end to end (rather than hop by hop).

In relation to reliable provisional responses, Cisco BroadWorks complies with the SIP standards documents concerning early media (*RFC 3261*) and offer/answer exchanges (RFC 3264). *RFC 6337* provides clarification on these interactions. BroadWorks supports the scenarios described in *RFC 6337*. These scenarios are covered in the call flows described in the next section.

Following are deviations on provisional response handling:

- When Cisco BroadWorks receives a PRACK request that does not match any unacknowledged provisional response, it returns a *200* response instead of a *481* response as required by *RFC 3262*.
- If Cisco BroadWorks times out while waiting for a PRACK request, it rejects the INVITE request with a *408 Request Timeout* response. This is different from the *RFC 3262* recommendation, which is to "reject the original request with a 5XX response".

3.16.2 Call Flows

Cisco BroadWorks supports the scenario in which the originating endpoint sends an offer SDP in the INVITE request and the terminating endpoint sends an answer SDP in a reliable provisional response. *RFC 6337* describes this scenario as the third pattern. The following call flow diagram shows the scenario.

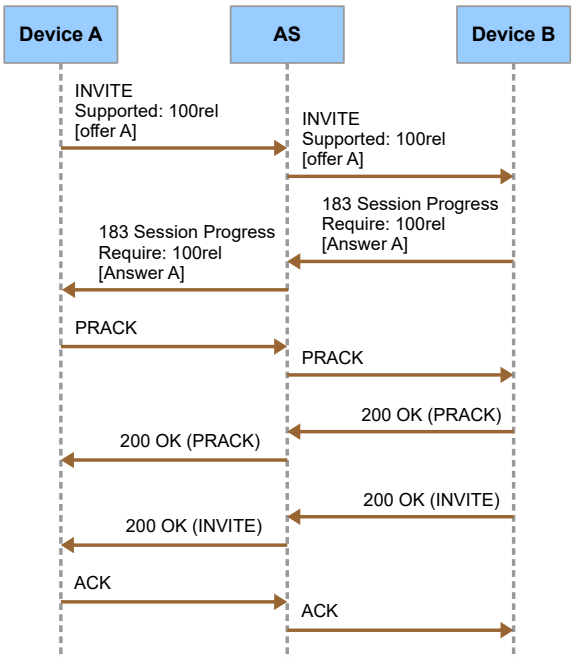

Figure 25 Offer/Answer with Answer in Reliable Provisional Response

Cisco BroadWorks also supports the scenario where the first reliable provisional response contains the offer SDP and the PRACK request contains the answer SDP. *RFC 6337* describes this scenario as the fourth pattern. If the originating endpoint does not send SDP in the initial INVITE request, then the terminating endpoint must send an offer SDP in the first reliable response. If the terminating endpoint sends a reliable provisional response, then it must send the offer SDP in that response. The originating endpoint must then send the answer SDP in the PRACK request. The following call flow diagram shows the scenario.

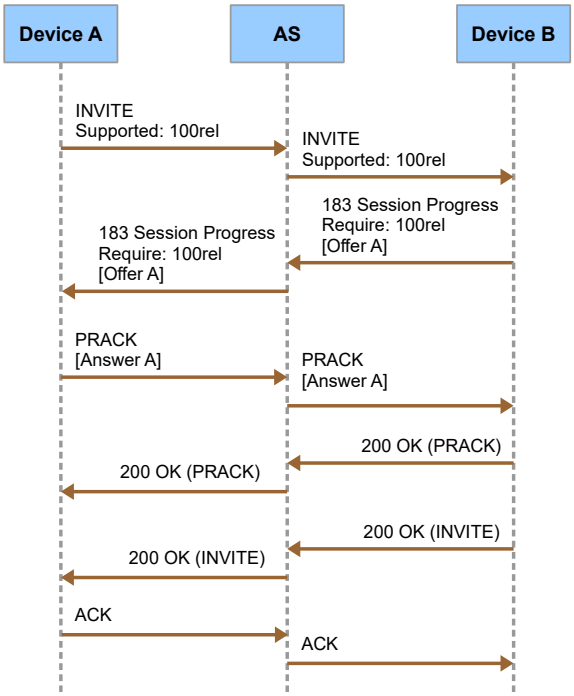

Figure 26 Offer/Answer with Offer in Reliable Provisional Response

RFC 6337 describes a fifth pattern, in which the originating endpoint sends a new offer SDP in the PRACK request and the terminating endpoint sends a new answer SDP in the 200 response to the PRACK. Cisco BroadWorks supports this scenario, which is shown in the following call flow diagram.

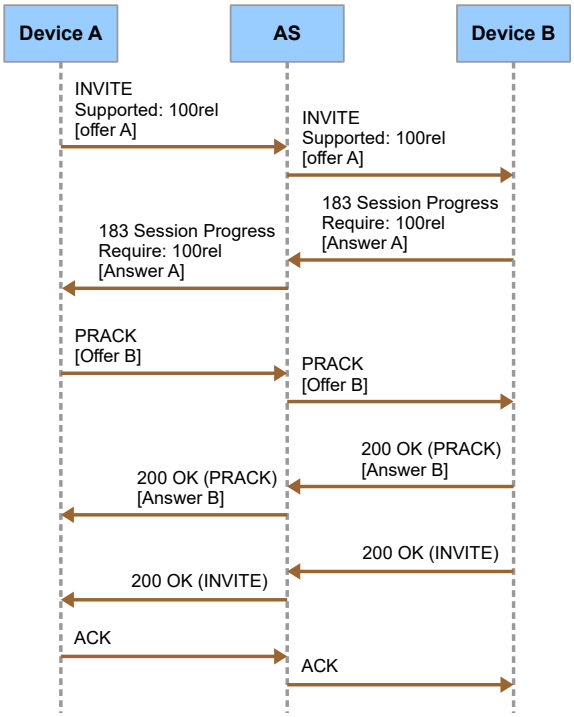

Figure 27 Offer/Answer with Second Offer in PRACK

3.17 Session Initiation Protocol UPDATE Method (RFC 3311)

Cisco BroadWorks fully supports this functionality.

When Cisco BroadWorks receives a SIP request or response from an endpoint, it remembers the methods in the *Allow* header. If an endpoint includes the UPDATE method in the *Allow* header, then Cisco BroadWorks assumes that endpoint supports UPDATE. Conversely, if the endpoint omits the UPDATE method from the *Allow* header, then Cisco BroadWorks assumes the endpoint does not support UPDATE and adapts its behavior accordingly.

Cisco BroadWorks supports UPDATE within an early dialog in order to initiate a new offer/answer exchange for early media. Cisco BroadWorks enforces the standard rules for offer/answer exchanges, and it may send an error response if an endpoint attempts to violate the rules. The calling endpoint may send a new offer SDP in an UPDATE request; however it may only do so after it receives an answer SDP in a reliable provisional response from the called endpoint. If the calling endpoint violates this rule and sends an UPDATE request with a new offer SDP before it receives an answer SDP in a reliable provisional response and sends the required PRACK request, then Cisco BroadWorks rejects the UPDATE request and sends a *500 Server internal error* response.

Within a confirmed dialog, if an endpoint device needs to initiate a new offer/answer SDP exchange, it should send a re-INVITE request rather than an UPDATE request. However, Cisco BroadWorks can receive and process received UPDATE requests with SDP on confirmed dialogs. If the destination endpoint device supports UPDATE (as indicated by UPDATE in the *Allow* header), then Cisco BroadWorks relays the received UPDATE request. If the destination endpoint does not support UPDATE, then Cisco BroadWorks sends to that endpoint an equivalent re-INVITE request.

> **NOTE**: *RFC 3311 section 5.1* recommends sending a re-INVITE instead of an UPDATE for confirmed dialogs.

The UPDATE method may be used to update other properties of the dialog. The following are some of the SIP headers that the Cisco BroadWorks Application Server allows to be updated by a received UPDATE request: *Allow, Contact, Min-SE, Session-Expires*, and *Supported*.

In addition, in the cases where a network device rejects an offer with a *488* response and includes a *Warning* header and/or a session descriptor, Cisco BroadWorks only proxies the *Warning* header back to the endpoint that generated the offer. Cisco BroadWorks ignores the Session Description included in the *488* response.

3.17.1 Call Flows

This section provides call flows that demonstrate how offer/answer via the UPDATE method is handled on Cisco BroadWorks:

- Two SIP early dialogs
- SIP early dialog and SIP established dialog
- Two SIP established dialogs

3.17.1.1 RFC 3311 - Two SIP Early Dialogs

The offer received from the network device in an UPDATE is transmitted to the other SIP endpoint in an UPDATE. The SIP endpoint provides the answer back to Cisco BroadWorks in the 200 UPDATE response. Cisco BroadWorks transmits the answer back to the network device in the 200 UPDATE response.

Either endpoint can initiate the offer/answer exchange using the UPDATE method.

- If a device rejects the offer by sending a *488* response, then Cisco BroadWorks sends a *488* response to the SIP endpoint that provided the offer and proxies the *Warning* header when appropriate. It is expected that the two SIP endpoints maintain their current connection attributes and that the call remains up and active.
- If a device does not support the UPDATE method, then Cisco BroadWorks sends a *500* response to the SIP endpoint that provided the offer. It is expected that the two SIP endpoints maintain their current connection attributes and that the call remains up and active.

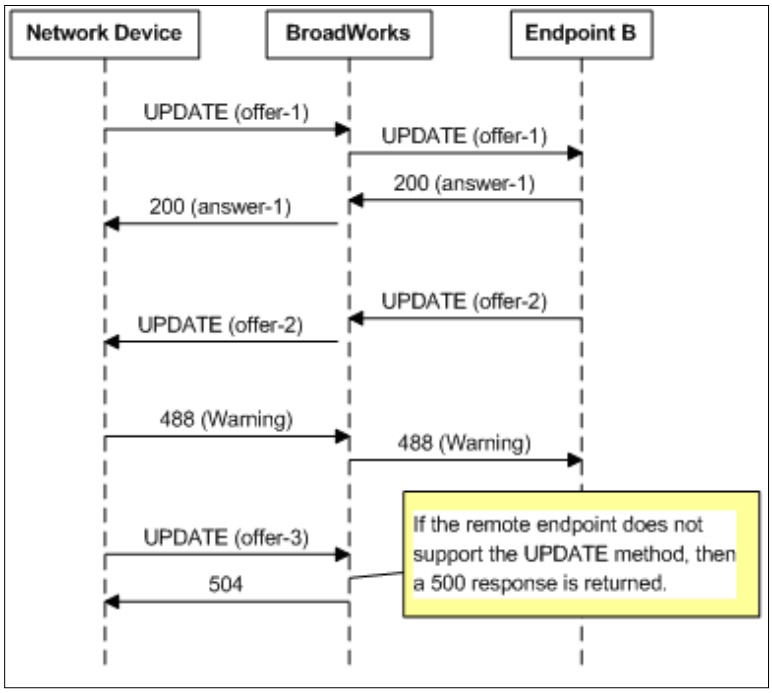

Figure 28 Two SIP Early Dialogs

3.17.1.2 RFC 3311 - SIP Established Dialog and SIP Early Dialog

In a call scenario involving a SIP endpoint (originating) with an established dialog, and a SIP endpoint (terminating) with an early dialog, the originating endpoint is either connected to Media Server ringback or early media provided by the terminating endpoint.

3.17.1.2.1 Offer from the SIP Endpoint with the Established Dialog

The offer received from a SIP endpoint in an established dialog (INVITE or UPDATE) is transmitted to the other SIP endpoint in an UPDATE (early dialog), if an early media stream is already established. The SIP endpoint provides the answer back to Cisco BroadWorks in the *200* UPDATE response. Cisco BroadWorks transmits the answer back to the other SIP endpoint in the *200* INVITE response.

- If the terminating endpoint rejects the offer, then a *488* response is sent back to the originating endpoint. It is expected that that the two SIP endpoints maintain their current connection attributes and that the call remains up and active.
- If the terminating endpoint does not support the UPDATE method or cannot be sent an offer (that is, answer provided in non-reliable *18x* response), then a *500* response is sent back to the originating endpoint. It is expected that that the two SIP endpoints maintain their current connection attributes and that the call remains up and active.

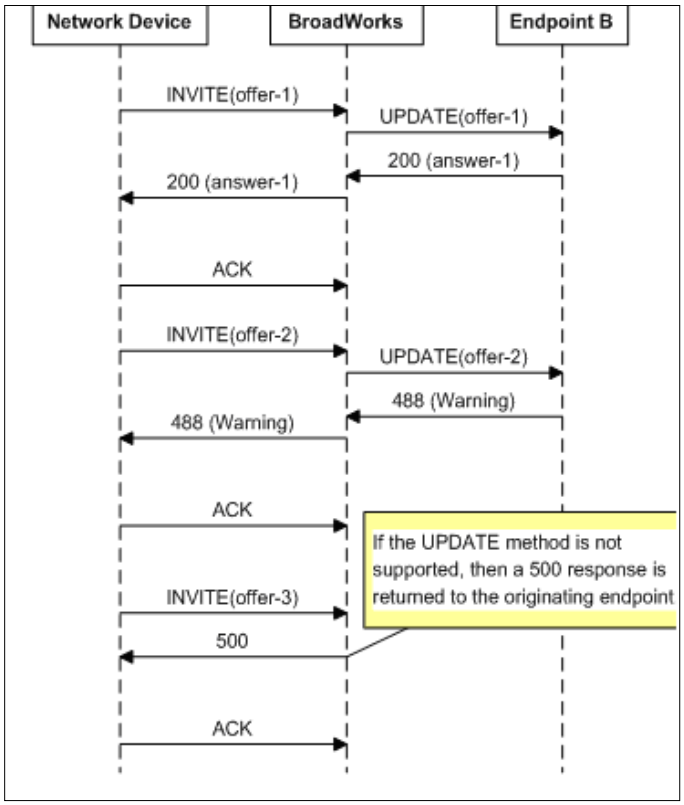

Figure 29 Offer from SIP Endpoint with Established Dialog

3.17.1.2.2 Offer from the SIP Endpoint with the Early Dialog

The offer received from a SIP endpoint in an UPDATE (early dialog) is transmitted to the other SIP endpoint in an INVITE or UPDATE (established dialog), depending on what the endpoint indicated support for (that is, *Allow* header).

- If the endpoint supports the UPDATE method, then the UPDATE method is used; otherwise, the INVITE method is used. The receiving SIP endpoint provides the answer back to Cisco BroadWorks in a *200* response. BroadWorks transmits the answer back to the other SIP endpoint in a 200 UPDATE response.
- If a SIP endpoint rejects the offer by sending a *488* response, then Cisco BroadWorks sends a *488* response to the SIP endpoint that provided the offer and proxies the *Warning* header when appropriate. It is expected that the two SIP endpoints maintain their current connection attributes and that the call remains up and active.

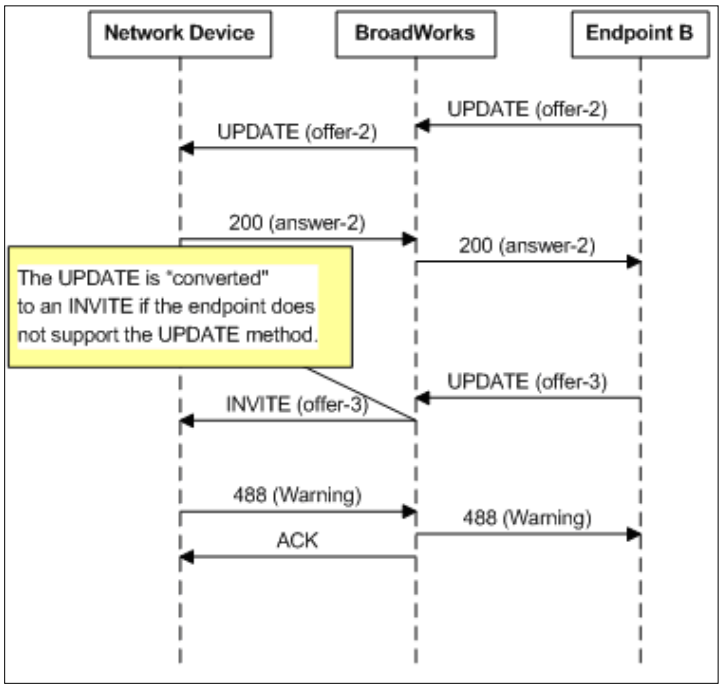

Figure 30 Offer from SIP Endpoint with Early Dialog

3.17.1.3 RFC 3311 - Two SIP Established Dialogs

The offer received from the SIP endpoint in an UPDATE (established dialog) is transmitted to the other SIP endpoint in an INVITE or UPDATE (also established dialog), depending on what the endpoint indicated support for (that is, *Allow* header).

- If the network device supports the UPDATE method, then the UPDATE method is used; otherwise, the INVITE method is used. The SIP endpoint that receives the offer provides the answer back to Cisco BroadWorks in a *200* response. Cisco BroadWorks transmits the answer back to the other SIP endpoint in a 200 UPDATE response.
- If a SIP endpoint rejects the offer by sending a *488* response, then Cisco BroadWorks sends a *488* response to the SIP endpoint that provided the offer and proxies the *Warning* header when appropriate. It is expected that the two SIP endpoints maintain their current connection attributes and that the call remains up and active.

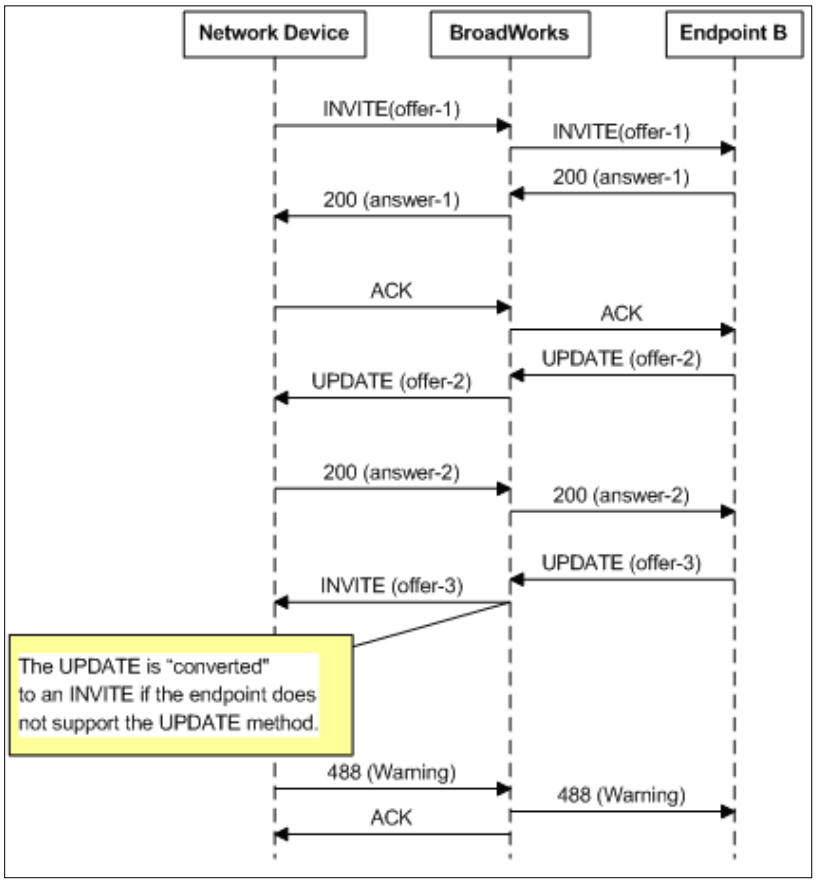

Figure 31 Two SIP Established Dialogs

3.18 Early Session Disposition Type for Session Initiation Protocol (RFC 3959)/Early Media and Ringing Tone Generation in Session Initiation Protocol (RFC 3960)

Cisco BroadWorks fully supports this functionality. It is strongly recommended that devices using the network interface on Cisco BroadWorks support this functionality.

RFC 3959 and *RFC 3960* define a method for providing early media in an independent session. The terminating endpoint provides an early-session offer in the *18x* response, and the originating endpoint provides the early-session answer in the PRACK. As such, beyond supporting early sessions, the two endpoints must also support reliable responses.

Endpoints that support or require early sessions must include the early-session option tag in the *Supported* or *Require* header of the INVITE request.

Cisco BroadWorks enables early session negotiation between two endpoints by “proxying” the early-session option tag from the originating endpoint to the terminating endpoint. In addition, if the terminating endpoint provides an early offer SDP, then this SDP is also proxied back to the originating endpoint. The following call flows show how the early-session option tag and the early-session SDPs are “proxied” across Cisco BroadWorks.

- If a call topology change occurs and the originating dialog is still in the alerting state, then the early-session option tag is proxied to the new terminating endpoint, such that an early-session can also be negotiated between the two endpoints. When this happens, the originating endpoint receives an 18x response with a new early-session offer and a different To-tag. The originating endpoint can then respond with a new early-session answer.
- If a call topology change occurs and the originating dialog is active (confirmed), then the early-session option tag is not proxied to the new terminating endpoint.

3.18.1 Call Flows

This section provides call flows that illustrate Cisco BroadWorks support of early-session handling:

- Early Session
- Early Session with Forking

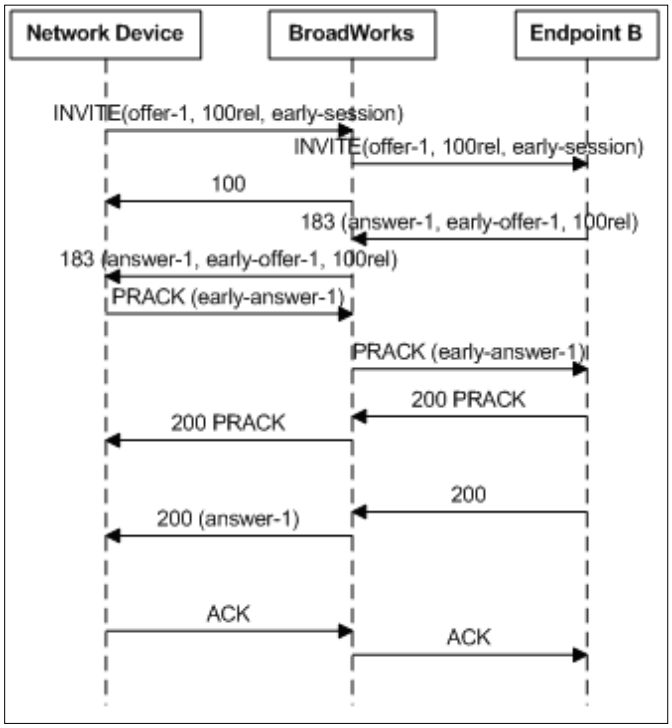

Figure 32 Early Session

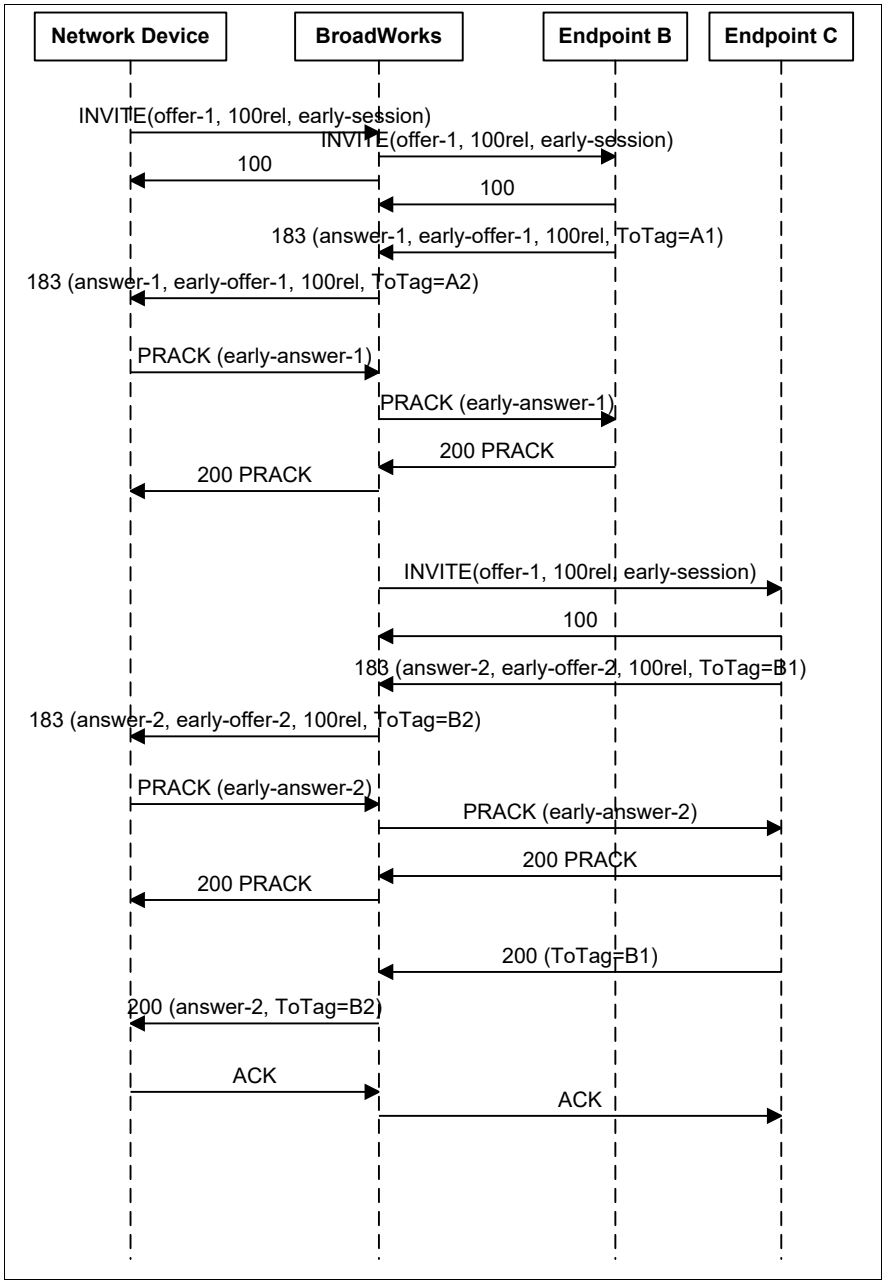

Figure 33 Early Session with Forking

3.19 Session Timers in Session Initiation Protocol (RFC 4028)

Cisco BroadWorks fully supports this functionality. Cisco BroadWorks uses the UPDATE method to refresh a session if the remote user agent indicates that it supports the UPDATE method via the *Allow* header. If UPDATE is not supported (or indicated) by the remote user agent, Cisco BroadWorks uses a re-INVITE to refresh the session. According to section 7.4 in *RFC 4028*, to determine if a session is still active, Cisco BroadWorks checks the origin line of the SDP of a re-INVITE to determine if the SDP has changed. If the origin line has not changed, Cisco BroadWorks treats the re-INVITE like a session extending/auditing re-INVITE.

Cisco BroadWorks does not require the session timer functionality because Cisco BroadWorks has its own session audit capability. However, both session timer and the Cisco BroadWorks session audit can be used simultaneously.

With the addition of this support, a new parameter has been added to enable/disable session timer support. The new parameter is *sessionTimer*. The default value is “false”. When the value is “true”, Cisco BroadWorks advertises support of the session timer functionality. When the value is “false”, Cisco BroadWorks does not advertise support of session timer.

By default, Cisco BroadWorks never includes the *Session-Expires* header in requests (including session refresh requests). Cisco BroadWorks only includes the *Session-Expires* header within *200* responses when the *sessionTimer* parameter is enabled and the request included the *Session-Expires* header. Cisco BroadWorks follows *RFC 4028* to determine the refresher for the session; however, the remote user agent always is preferred by Cisco BroadWorks and chosen when possible.

Cisco BroadWorks can also be configured to explicitly request the SIP session timer when the device (or a proxy) supports it. This means that if the Application Server receives an INVITE request with the timer option in the *Supported* header, but no *Session-Expires* header, it sends the timer option in the *Require* header and the *Session-Expires* header in the *200 OK* response. In addition, when the Application Server sends an INVITE request, it includes the *Session-Expires* header directly. If the target device or an intervening proxy does not support this option, the *200 OK* response simply does not contain the *Session-Expires* header.

In both of these scenarios, the outgoing *Session-Expires* header contains the configured preferred session timer value (from *AS_CLI/System/CallP/SessionAudit>sipSessionExpiresTimer*) and sets the refresher to the configured value (the new configurable parameter is *AS_CLI/System/CallP/SessionAudit> preferredSessionTimerRefresher*). When the *preferredSessionTimerRefresher* is set to “local”, the Application Server sets the refresher parameter so that it controls the refreshes (that is, uas in *200 OK* responses and uac in INVITE requests). When this parameter is set to “remote”, the Application Server tries to get the far-end to handle the refreshes (by setting the refresher parameter to “uac” in *200 OK* responses and “uas” in INVITE requests).

Additionally, the minimum allowed value for the *sessionExpiresMinimum* is changed from “0” to “30”.

3.20 Locating SIP Servers (RFC 3263)

Cisco BroadWorks supports this functionality. When TCP is enabled on Cisco BroadWorks, Cisco BroadWorks uses NAPTR lookups to determine the appropriate transport for the URI unless the contact specifies the transport. When TCP is used, Cisco BroadWorks includes the *transport=tcp* parameter in the Contact entry. Explicitly specifying the transport provides better interoperability with devices unable to perform NAPTR or SRV queries to recognize that Cisco BroadWorks prefers the continued use of TCP for the dialog. For additional information on *RFC 3263* support, see section *3.1.2.1 Differences between UDP and TCP Transports for SIP*.

3.20.1 DNS Query Procedure

Cisco BroadWorks supports NAPTR, SRV, AAAA, and A records in accordance with *RFC 3263*.

When Cisco BroadWorks has a SIP URI as the target for a SIP request, it performs a sequence of steps to select the transport protocol, IP address, and port number for sending the request. These steps include DNS queries in accordance with *RFC 3263* and are depicted in the annotated diagrams that follow. The processing logic depends on several factors, including Cisco BroadWorks configuration and the information in the SIP URI.

If conditions allow it, Cisco BroadWorks first performs a NAPTR query, as shown in the following diagram.

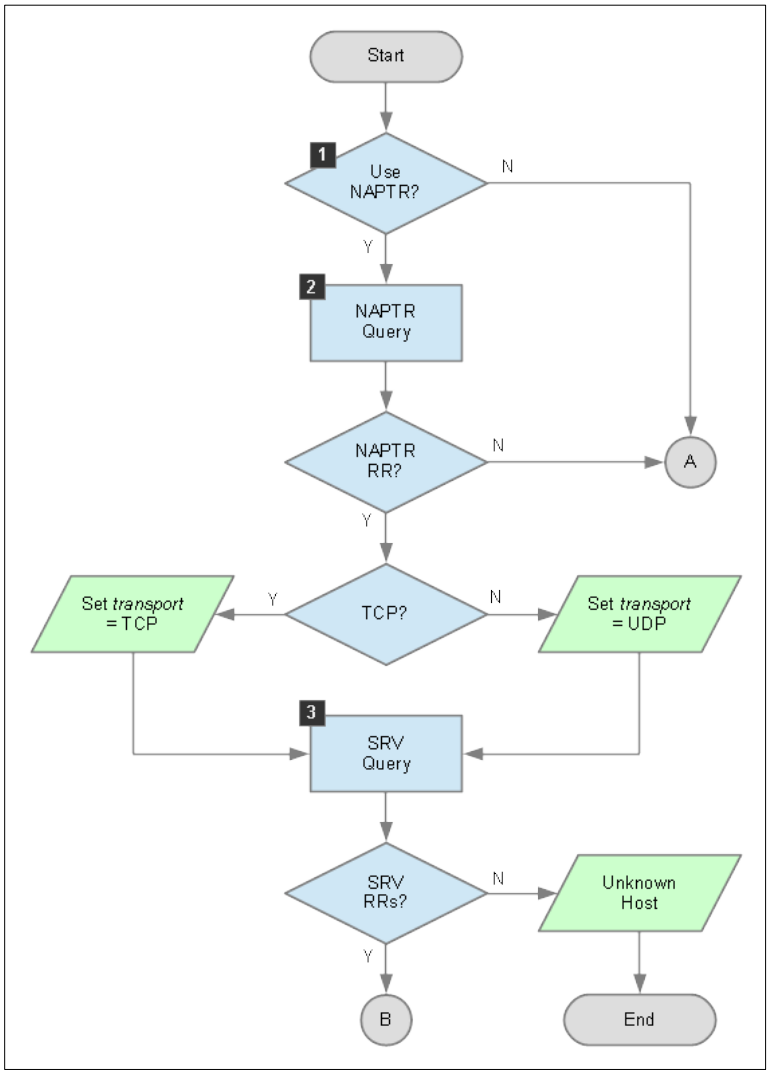

Figure 34 Flow Diagram for DNS NAPTR Query

The following notes explain the callouts in the diagram.

1) Cisco BroadWorks performs a NAPTR query if all of the following conditions are true:
   - The SIP parameter *supportTcp* is set to “true”.
   - The SIP parameter *supportDnsNaptr* is set to “true”.
   - The URI does not have a *transport* parameter.
   - The URI does not contain a port number.
   - The URI host is a domain name (that is, the host is not an IPv4 address or an IPv6 address).

2) If Cisco BroadWorks receives NAPTR records, it processes them to select a single preferred record. The following points provide the details of the procedure.

   – Cisco BroadWorks screens all records and accepts a record only if the following conditions are all true:

      – The service field contains “SIP+D2T” (indicating TCP) or “SIP+D2U” (indicating UDP).

      – The flag field contains “S” (indicating SRV)[2].

   – After screening all records, Cisco BroadWorks selects a single preferred record to process further. Cisco BroadWorks applies the following ordering criteria when comparing two records:

      – If the order value is different, then Cisco BroadWorks prefers the record with the lowest order value.

      – If the order value is the same but the preference value is different, then Cisco BroadWorks prefers the record with the lowest preference value.

      – If the order value and the preference value are the same but the service field is different, then Cisco BroadWorks prefers the record with service field set to “SIP+D2T”. (That is, Cisco BroadWorks prefers TCP.)

      – If the order, preference, and service fields are the same, then Cisco BroadWorks makes a random choice.

3) After Cisco BroadWorks selects a single, usable NAPTR record, it performs a SRV query using the replacement field value as the domain name.

[2] The Application Server can process a NAPTR record that contains the “A” flag. However, this special case is not described in *RFC 3263* is beyond the scope of this document.

If Cisco BroadWorks has no NAPTR records, either because the DNS server did not send any acceptable records or because Cisco BroadWorks did not query for them, then it may perform an SRV query, as shown in the following figure.

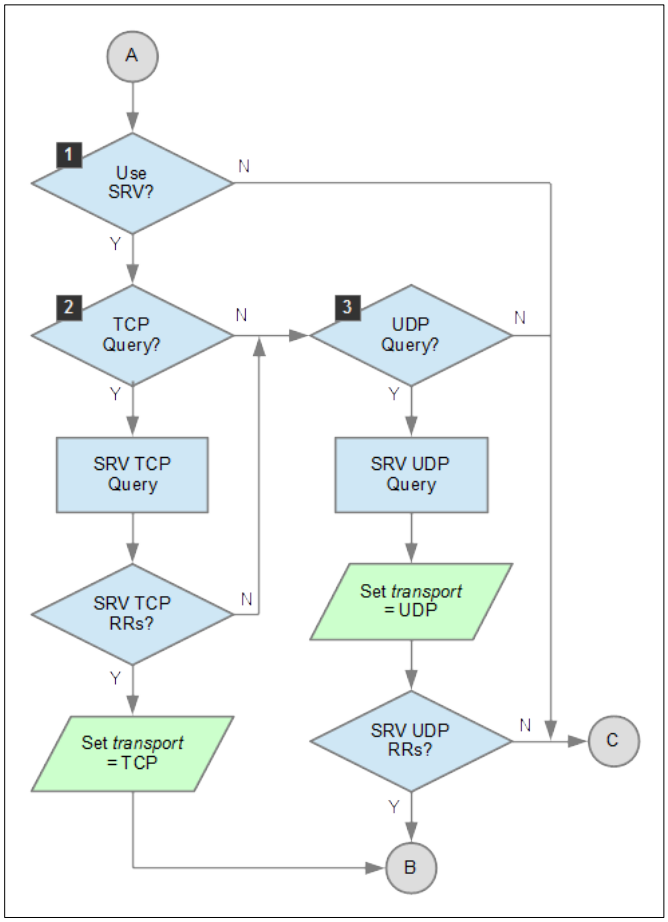

Figure 35 Flow Diagram for DNS SRV Query

The following notes explain the callouts in the previous diagram.

1) Cisco BroadWorks performs an SRV query if all of the following conditions are true:
   - The SIP parameter *supportDnsSrv* is set to “true”.
   - The URI does not contain a port.
   - The URI host is a domain name (that is, the host is not an IPv4 address or an IPv6 address).
2) Cisco BroadWorks performs a TCP SRV query if both of the following conditions are true:
   - The SIP parameter *supportTcp* is set to “true”.
   - The URI has no *transport* parameter or it has *transport=TCP*.
3) Cisco BroadWorks performs a UDP SRV query if both of the following conditions are true:
   - Cisco BroadWorks has no TCP SRV records, either because the DNS did not return any records or because Cisco BroadWorks did not query for them.

- The URI has no *transport* parameter or it has *transport=UDP*.

If Cisco BroadWorks receives SRV records, then it processes them as shown in the following diagram.

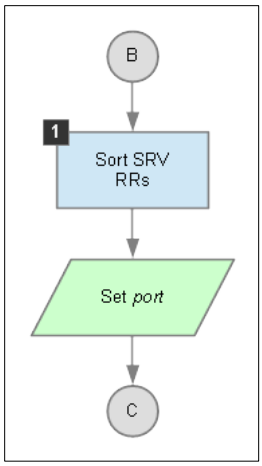

Figure 36 Flow Diagram for DNS SRV Record Processing

The following note explains the callouts in the diagram.

1) Cisco BroadWorks sorts the records in accordance with *RFC 2782*. The following points provide the details of the procedure.

   - If records have different priority values, then Cisco BroadWorks prefers the record with the lowest value.
   - If several records have the same priority value, then Cisco BroadWorks orders those records randomly using weighted probabilities provided in the records.

   Cisco BroadWorks final processing steps may include DNS A queries or DNS AAAA queries. Unlike the NAPTR and SRV queries, the A and AAAA queries do not resolve the transport protocol or the port number. Therefore, Cisco BroadWorks performs the steps shown in the following diagram before performing an A query or AAAA query.

If Cisco BroadWorks performed NAPTR and/or SRV queries earlier, then it may have an ordered list of domain names. In such case, it performs the following processing on all domain names and generates a list of IP addresses.

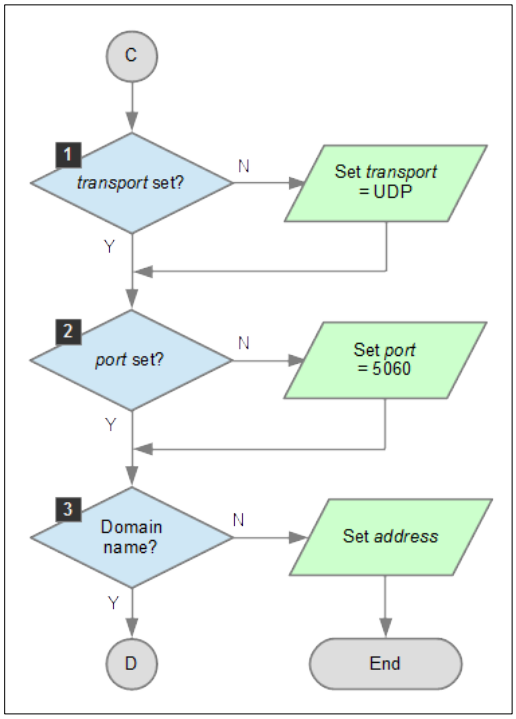

Figure 37 Flow Diagram for DNS A/AAAA Query Preparation

The following notes explain the callouts in the diagram.

1) At this point in the processing, the transport protocol may already be selected, either from a *transport* parameter in the URI or as the result of the earlier processing steps (for example, the NAPTR query). If it is not selected, then Cisco BroadWorks selects UDP as the default.

2) At this point in the processing, the port number may already be selected, either from a value in the URI or as the result of the earlier processing steps (for example, the SRV query). If it is not selected, then Cisco BroadWorks selects 5060 as the default.

3) The host part of the URI may be an IPv4 address or an IPv6 address, in which case Cisco BroadWorks uses the explicit address.

In the final step, Cisco BroadWorks queries the DNS for A or AAAA records. The flow for this procedure is shown in the following diagram.

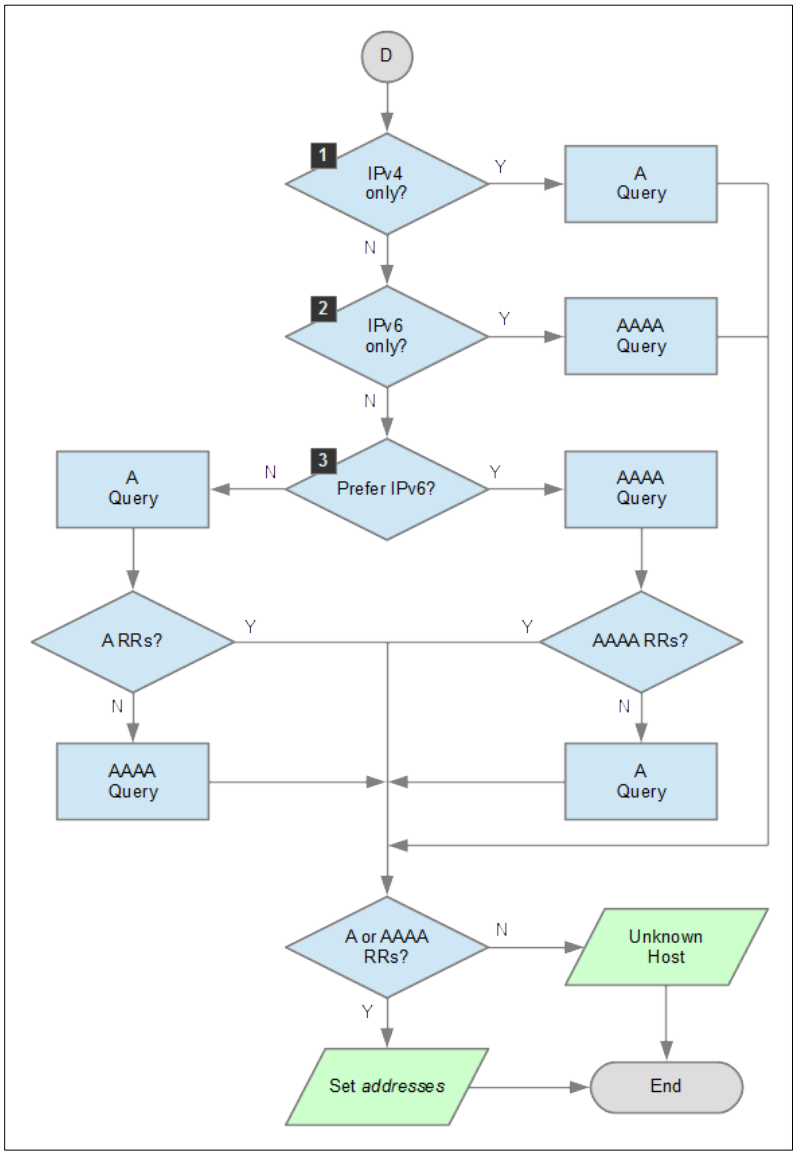

Figure 38 Flow Diagram for DNS A/AAAA Query

The following notes explain the callouts in the diagram.

1) If the SIP parameter *sipIpVersion* is set to “ipv4”, then Cisco BroadWorks queries for A records only.

2) If the SIP parameter *sipIpVersion* is set to “ipv6”, then Cisco BroadWorks queries for AAAA records only.

3) If the SIP parameter *sipIpVersion* is set to “both”, then Cisco BroadWorks queries for either A or AAAA records. Cisco BroadWorks prefers IPv6 when both of the following conditions are true:

   – *java.net.preferIPv4Stack* is set to “false” (that is, *java.net.preferIPv4Stack* is not set at all or is set to a value other than “true”)

– java.net.preferIPv6Addresses is set to “true”

An administrator can set the value of *java.net.preferIPv4Stack* from the CLI using the set command at the */System/GeneralSettings* level.

```
AS_CLI/System/GeneralSettings> set preferIPv4Stack false
...Done
```

An administrator can set the value of *java.net.preferIPv6Addresses* as a container option from the CLI level */Maintenance/ContainerOptions*.

```
AS_CLI/Maintenance/ContainerOptions> add execution
java.net.preferIPv6Addresses true
*** Warning: Broadworks needs to be restarted for the changes to take
effect ***
```

3.21 Best Current Practices for Third Party Call Control (3PCC) in SIP (RFC 3725)

Cisco BroadWorks implements third-party call control using a back-to-back user agent (B2BUA). Network devices must adhere to section 11 of *RFC 3725* to facilitate advanced services.

3.22 SIP/PSTN Interworking

3.22.1 SIP for Telephones (SIP-T): Context and Architectures (RFC 3372)/Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol Mapping (RFC 3398)

Cisco BroadWorks does not fully support this functionality. Specifically, Cisco BroadWorks does not generate the INFO request as described in this RFC. If Cisco BroadWorks receives an INFO request for a SIP-T event, Cisco BroadWorks ignores the request unless configured in the Cisco BroadWorks content-type list. For information on Cisco BroadWorks handling of content-types, see section *3.29.1 Cisco BroadWorks Content-type Support*.

Where applicable, Cisco BroadWorks uses *RFC 3398* for equivalent telephony functions related to a line-side device, including message mapping and negative response codes.

Cisco BroadWorks also supports early media transitions via *RFC 3398* to allow receipt of a *180 Ringing* without SDP, for a CPG received by the network device, after receiving a *183 Session Progress* with SDP associated with the ACM received by the network device. Prior to this behavior, Cisco BroadWorks inserted ringback via the Media Server causing garbled ringback.

3.23 SIP-specific Event Notification (RFC 6665)

Cisco BroadWorks fully supports this functionality. This functionality provides Cisco BroadWorks with a powerful service creation platform that can be extended by adding support for additional event packages. Cisco BroadWorks supports the following event packages on the network interface:

- Calling-name (Cisco BroadWorks proprietary event package for calling name delivery). For more information, see the *Cisco BroadWorks Calling Name Interface Specification* [36].
- Message-summary (message-summary).

Support for additional event packages will be added to Cisco BroadWorks in the future, providing third-party access to Cisco BroadWorks subscriber data.

Cisco BroadWorks currently does not support the ability to share a dialog with calls and subscriptions. Additionally, Cisco BroadWorks does not support multiple subscriptions on a dialog at the same time.

3.24 SIP INFO Method (RFC 2976, RFC 6086)

Cisco BroadWorks supports this functionality. When the INFO request contains a message body configured with a configured Content-Type, Cisco BroadWorks proxies the INFO request to the remote party.

For example, applications that support video codecs such as H.263 need to convey media control information between participating devices. While it is possible to convey such information directly in the media streams themselves, certain information is considered closely related to the application logic. This higher-level application information is best conveyed through the signaling channel, rather than through the media stream. In the case where the applications use SIP signaling, such information is conveyed in SIP INFO requests.

For more information on message body proxy capabilities, see section *3.29.1 Cisco BroadWorks Content-type Support*.

In addition to proxying INFO requests, Cisco BroadWorks processes some specific INFO content. Cisco BroadWorks uses the INFO request to support flash-based services on access devices such as SIP gateways, which provide a SIP interface for analog (FXS) lines. Cisco BroadWorks also recognizes DTMF signals conveyed by INFO requests in the following formats: application/dtmf-relay, application/dtmf, and audio/telephone-event.

Cisco BroadWorks performs the following handling of INFO request based on the content-type:

Body Type	Behavior
Application/ media_control+xml	Proxied if configured under *AS_CLI/Interface/SIP/ContentType*. It is configured by default on install and upgrades.
Application/dtmf	If the content is "#" and the system parameter *treatDTMFPoundAsFlash* is set, interpret as a Flash. *treatDTMFPoundAsFlash* is enabled by default on install and upgrades. If configured under *AS_CLI/Interface/SIP/ContentType*, it is proxied as well. This body type is not configured by default on installation and upgrades.
Application/broadsoft	Interpreted and never proxied.
Others	Proxied if configured under *AS_CLI/Interface/SIP/ContentType*.

Table 1 Special INFO Requests

3.24.1 Video Support via INFO Request

Cisco BroadWorks supports the ability to proxy INFO requests after a call has been established that convey media control information for video calls. The INFO requests convey the media control information in the request body, which has the MIME type application/media_control+xml.

The Application Server sends the request with end-to-end reliability. In other words, the Application Server sends back a 200 response to the INFO request after it receives a 200 response from the forwarded INFO request.

The specific media control information that the INFO request conveys includes:

- Video Picture Fast Update Request (decoder to encoder)
- Video GOB Fast Update Request (decoder to encoder)
- Video MB Fast Update Request (decoder to encoder)
- Video Picture Freeze Request (encoder to decoder)

The H.263 standard provides more information about these requests. The Internet draft entitled "XML Schema for Media Control" describes the encoding of this media control information in an XML payload with MIME type application/media_control+xml. For more information, see the *XML Schema for Media Control* [38].

Figure 39 shows one possible scenario in which the Application Server must proxy SIP INFO requests between devices. Audio and video streams are established between a video phone and a gateway. Signaling is handled by the Application Server via SIP messages. When the gateway needs to convey media control information to the video phone, it sends a SIP INFO request to the Application Server, which forwards it to the video phone. Similarly, when the video phone needs to convey media control information to the gateway, it sends a SIP INFO request to the Application Server, which forwards it to the gateway.

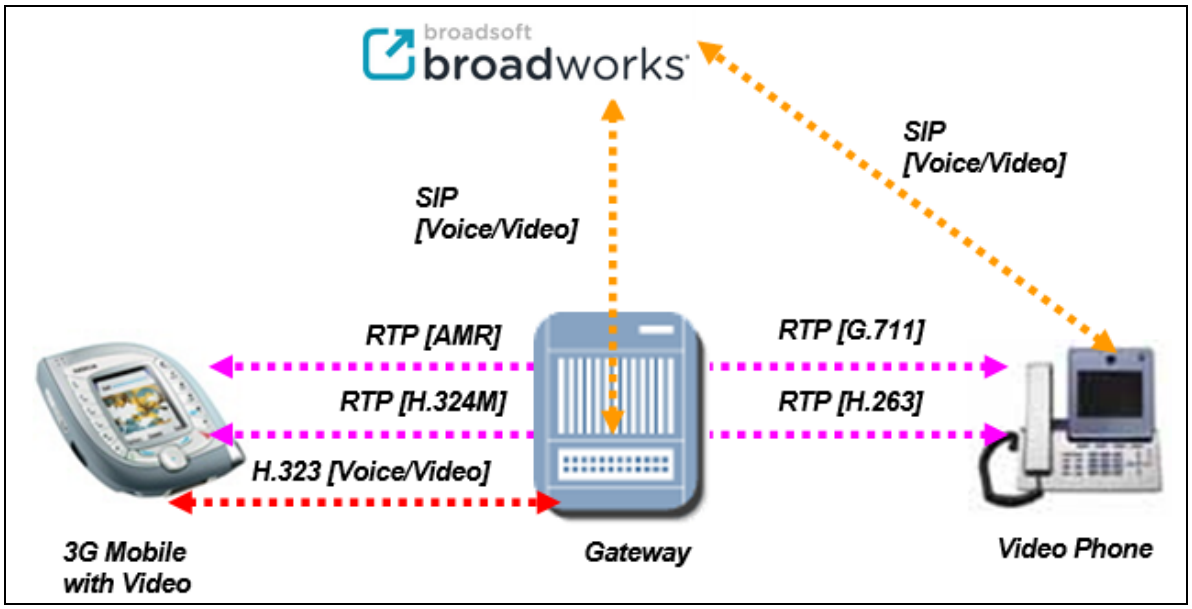

Figure 39 Cisco BroadWorks Support for INFO Proxying for Media Control with Video Applications

If the Application Server receives an INFO request before the call is answered, it does not proxy that request.

The SIP device must advertise in an *Allow* header field that it accepts the INFO request. Otherwise, the Application Server does not send that device to the INFO request.

The Media Server does not support receiving INFO requests for the proxying of media control for video applications. The Media Server is not a video encoder or decoder; it is a video streaming server. Apple QuickTime Pro typically performs the video encoding before a file is transmitted to a Media Server. For that reason, the Media Server cannot act on video control messages, and thus ignores them. The Media Server does not send SIP INFO requests with video control messages.

3.24.2 DTMF Support via the INFO Request

BroadWorks supports out-of-band DTMF through the INFO request. The following Content-Types carrying DTMF are supported:

- application/dtmf-relay
- application/dtmf
- audio/telephone-event

Assuming the corresponding Content-Type is configured, these messages are proxied when two parties are connected. In addition, Cisco BroadWorks Media Server recognizes the DTMF signals when digit extraction is required. The following rules apply:

- A single DTMF digit must appear in each INFO request.
- The Media Server recognizes *application/dtmf-relay* message bodies. *application/dtmf* and audio/telephone-event are transparently converted to *application/dtmf-relay* by the Application Server before proxying them to the Media Server.
- If a single DTMF event is sent simultaneously over the RTP media stream and in an INFO request to the Media Server, the Media Server reports two DTMF events. This, in fact, would cause each DTMF event to appear repeated. It is imperative that a device sending a DTMF event using an INFO request not send the same DTMF event over the RTP media stream (not modulated in the voice path or through *RFC 2833*).

shows a situation in which a user accesses their voice portal. The Application Server is configured to proxy the application/dtmf-relay to the Media Server.

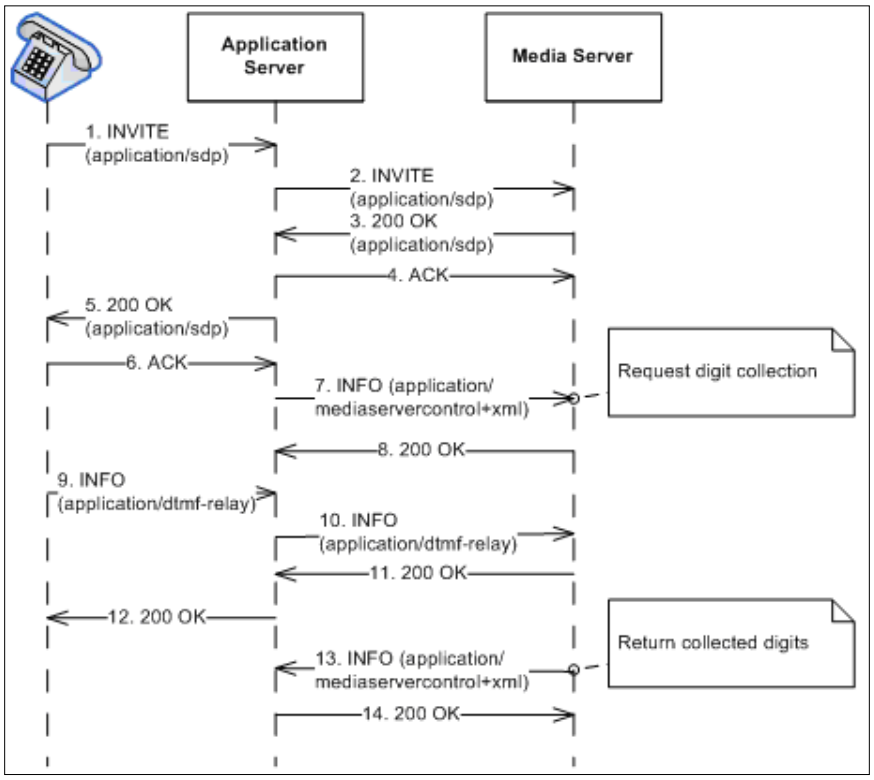

Figure 40 Media Server Processes Application/dtmf-relay

3.24.3 Transfer Notification for BroadWorks Mobility

Cisco BroadWorks uses the info package *x-broadworks-transfer-notification* to provide call transfer notification between two Application Servers. In particular, Cisco BroadWorks uses this info package to support transfer notification when a service provider has built a multiple persona solution using the BroadWorks Anywhere and BroadWorks Mobility services. The scenario is depicted in the following call flow diagram.

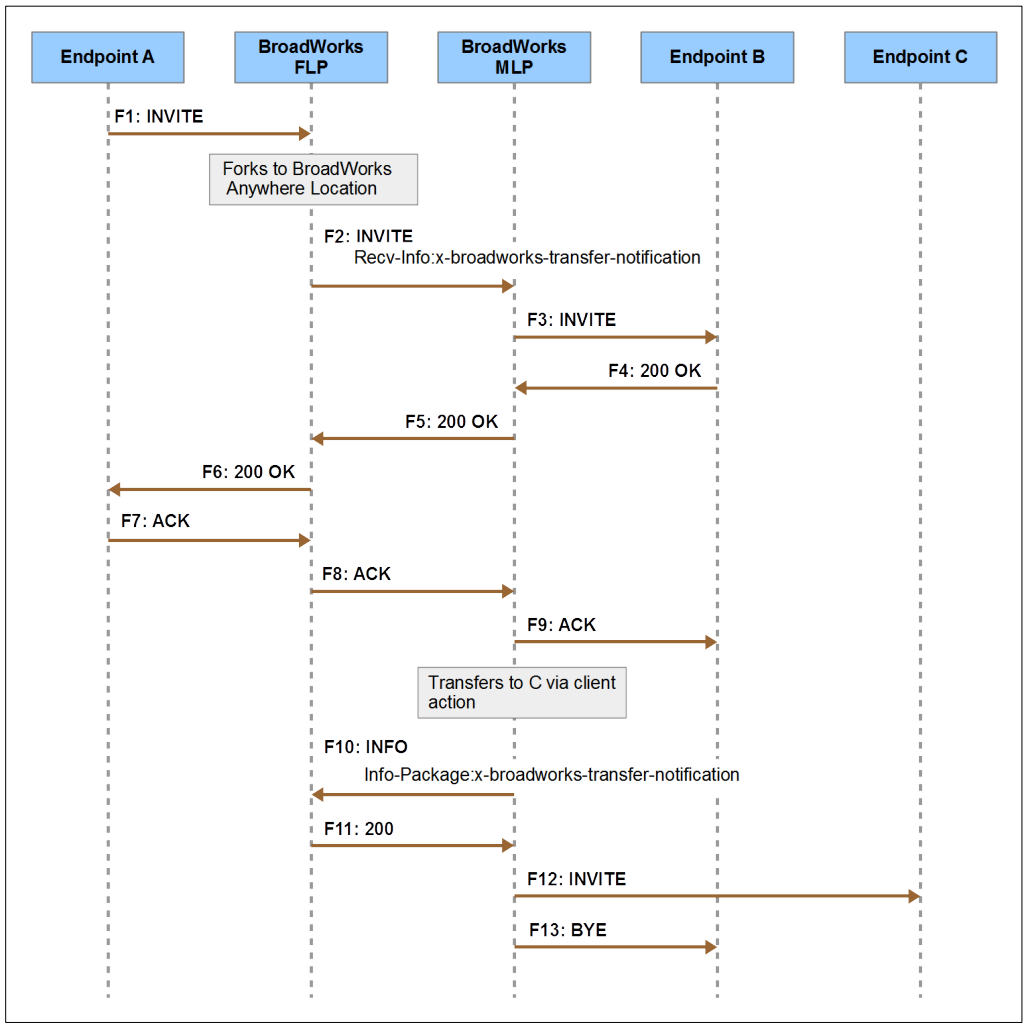

Figure 41 Transfer Notification for BroadWorks Mobility

The scenario in the diagram centers on Cisco BroadWorks User B. In order to support multiple personas, User B is configured with two service profiles on Cisco BroadWorks. User B has a fixed line persona (FLP) profile, which is configured with a BroadWorks Anywhere location that terminates to User B's mobile line persona (MLP) profile.

As shown in the diagram, the scenario begins with User A, at Endpoint A, calling Cisco BroadWorks User B's fixed-line number. Cisco BroadWorks handles the incoming INVITE request using User B's FLP. Cisco BroadWorks executes the BroadWorks Anywhere service and forks the INVITE request to User B's mobility number. Cisco BroadWorks handles this INVITE request using User B's MLP and routes the call toward User B's mobile device. The INVITE request from the FLP to the MLP has a *Recv-Info* header that contains the value *x-broadworks-transfer-notification*, which indicates that the FLP wishes to receive an INFO request for the *x-broadworks-transfer-notification* info package in the event of a call transfer.

After answer, User B uses a client application to request a transfer to User C. (Note that this example describes a client-initiated transfer. Other transfer scenarios are supported as well.) Before the MLP starts the transfer, it sends an INFO request to the FLP for the *x-broadworks-transfer-notification* info package. When the FLP receives this INFO request, it takes the appropriate actions for a transferred call.

3.25 Message Summary and MWI Event Package for SIP (RFC 3842)

Cisco BroadWorks supports this functionality to deliver Voice Message Waiting Indications (MWI) from Third-Party Voice Mail systems to Cisco BroadWorks subscribers.

Cisco BroadWorks does not support the SUBSCRIBE portion of this RFC. However, Cisco BroadWorks accepts NOTIFY requests on the network interface for Cisco BroadWorks subscribers, based on the network interface access control list. Cisco BroadWorks rejects NOTIFY requests with the application/simple-message-summary body, which are not destined for a Cisco BroadWorks subscriber.

A NOTIFY request for the message-summary event package may be redirected by Cisco BroadWorks. Typically, the Third-Party Voice Mail system is configured with the address of the Cisco BroadWorks Network Server cluster. The NOTIFY request is sent to this cluster where the Network Server redirects the NOTIFY to the appropriate Application Server (that is, the Application Server currently hosting the subscriber).

Although Cisco BroadWorks may receive the full message-summary content, Cisco BroadWorks only provides the Messages-Waiting and Voice-Message portion to the Cisco BroadWorks subscriber.

Cisco BroadWorks only sends the status-line (for example, Messages-Waiting) and the voice-message message-context-class (for example, Voice-Message) portion of the application/simple-message-summary body. Cisco BroadWorks sends a NOTIFY for every message status change to accurately update the "new" message count.

Cisco BroadWorks (optionally) sends the number of saved and urgent messages in addition to new messages. This is configured from the CLI as follows.

```
AS_CLI/Service/VoiceMessageSummaryUpdate> set
sendSavedAndUrgentMWIOnNotification true
```

3.26 SIP Extension for Instant Messaging (RFC 3428)

Cisco BroadWorks supports Short Message Service (SMS) handling using MESSAGE request according to *RFC 3428* [59]. This service allows text messages to be sent from:

- Public Switched Telephone Network (PSTN) to Cisco BroadWorks subscribers,
- Cisco BroadWorks subscribers to Cisco BroadWorks subscribers, and
- Cisco BroadWorks subscribers to the PSTN.

When a short message originates from or terminates to the PSTN, a Short Message Service Center (SMSC) transforms the SIP MESSAGE request from and to other protocols as needed.

The Application Server supports outgoing short messages originated from Cisco BroadWorks subscribers by accepting a MESSAGE request from the access device. For an origination, the Application Server interacts with the Network Server to determine how to route the MESSAGE request to the proper SMSC.

The following figure provides an example of a message flow for an SMS origination from a Cisco BroadWorks subscriber.

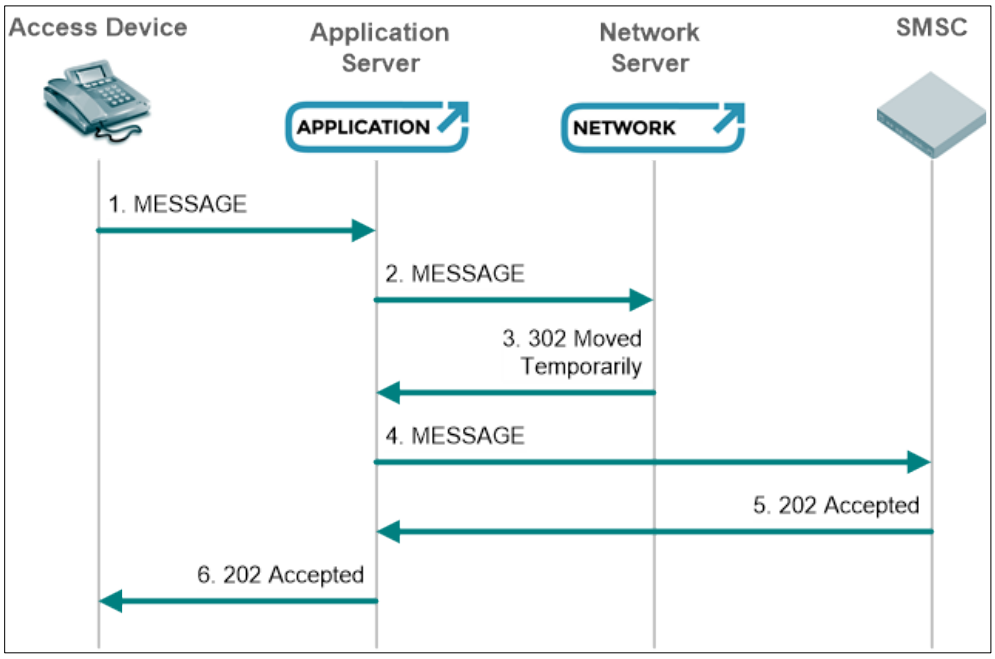

Figure 42 SMS Origination from Cisco BroadWorks Subscriber

The following figure provides an example of a message flow for an SMS termination to a Cisco BroadWorks subscriber.

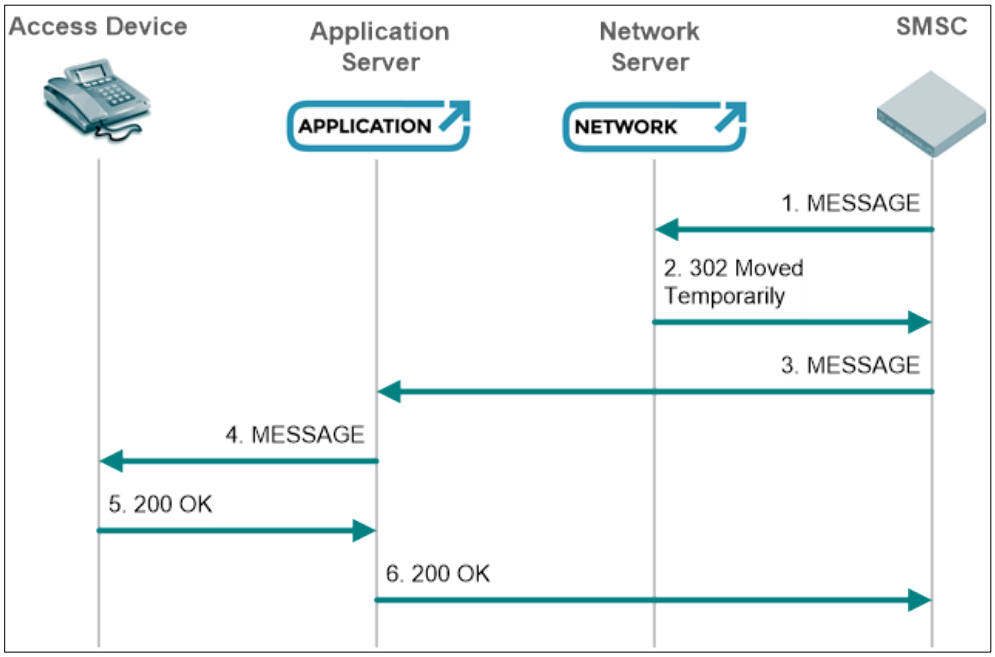

Figure 43 SMS Termination to Cisco BroadWorks Subscriber

NOTE 1: Cisco BroadWorks does not support receiving a MESSAGE request within a dialog.

NOTE 2: The Application Server does not send the 100 response and does not adjust the retry timer upon receiving the 100 response.

The following is a MESSAGE example containing a simple text message as a *text/plain* media type in the message body.

```
MESSAGE sip:3015551111@smsc.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.40.10;branch=z9hG4bKBroadWorks.-1t1hjce-
192.168.40.11V5060-0-929489740-946677141-1240957044390-
From:"example user"<sip:+13015550000@broadsoft.com;user=phone>;
tag=946677141-1240957044390-
To:<sip:3015551111@smsc.com;user=phone>
Call-ID:BW131909796040500-565063280@as.broadsoft.com
CSeq:929489740 MESSAGE
P-Asserted-Identity:"example user"
<sip:+13015550000@broadsoft.com;user=phone>
Privacy:none
Max-Forwards:10
Content-Type:text/plain;charset=iso-8859-1
Content-Length:12

Hello World!
```

3.27 RTP Payload for DTMF Digits (RFC 4733)

Cisco BroadWorks supports this specification. The Cisco BroadWorks Media Server uses *RFC 4733* to provide the ability to collect dual-tone multi-frequency (DTMF) digits for Interactive Voice Response (IVR) services. If the device connecting to the Media Server supports *RFC 4733*, as indicated by the telephone-event media format in the SDP, the Cisco BroadWorks Media Server uses *RFC 4733* to collect DTMF digits from the device. Otherwise, the Cisco BroadWorks Media Server collects the digits via the decoding of the audio waveform RTP packets.

Network devices should support *RFC 4733* as it guarantees reliable delivery of DTMF digits from the network device to the entity collecting the digits. The use of telephone-event media is especially important when waveform audio is carried in a compressed format, such as G.729, since DTMF cannot be reliably transmitted in compressed audio.

3.28 SIP Support for Real-Time Fax: Call Flow Examples and Best Current Practices (T.38 Annex D)

Cisco BroadWorks supports T.38 fax, as described in T.38 Annex D *SIP/SDP call establishment procedures*. Cisco BroadWorks supports both receiving and sending a fax. Cisco BroadWorks supports the UDPTL transport for T.38 data transfer and negotiates the fax session parameters on a per-call basis. Cisco BroadWorks supports the TIFF-FX specification profile F in *RFC 3949*, *File Format for Internet Fax*.

3.28.1 Fax Reception

Incoming fax calls terminate to a dedicated fax number associated with a Cisco BroadWorks subscriber. Cisco BroadWorks supports incoming calls that include both the audio codec and the fax (udptl) codec in the SDP in the INVITE, and calls that only include the audio codec and negotiate dynamically to T38.

The “Receive” mode of the T.30 protocol is supported. The “Poll” and “Turnaround Poll” modes are not supported.

3.28.1.1 Message Sequence Diagrams

A successful “faxrecord” session is depicted in *Figure 44*.

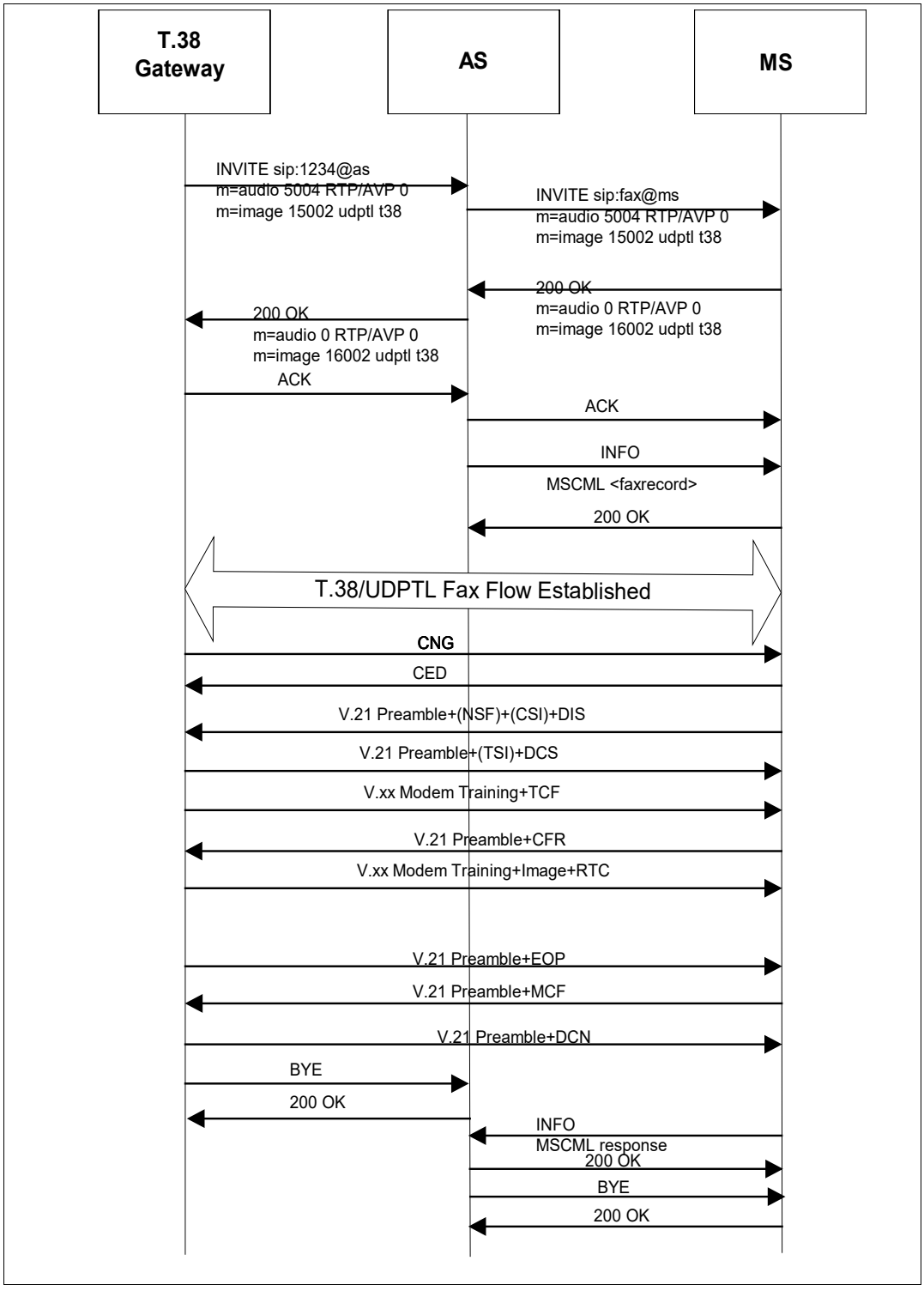

Figure 44 Example of Successful “faxrecord” Session

Figure 45 shows a basic fax call flow (Re-invite scenario) between the emitting T.38 fax gateway, Application Server, and Media Server. The Re-invite scenario involves initiating the media session as the audio session, and later the Media Server re-invites for a fax session.

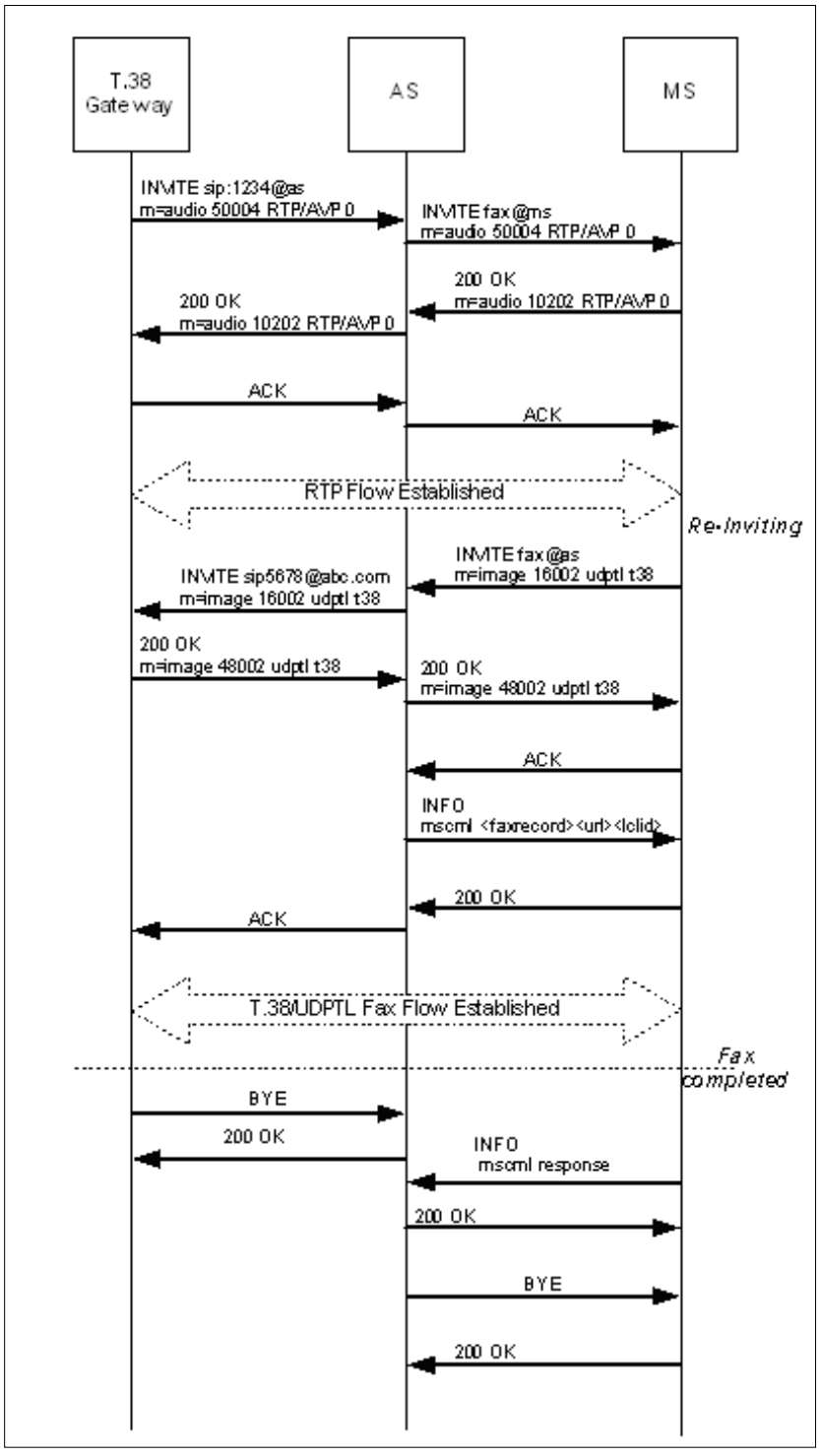

Figure 45 Successful <faxrecord> Call Flow in Re-invite Scenario

Figure 46 shows an unsuccessful “faxrecord” session.

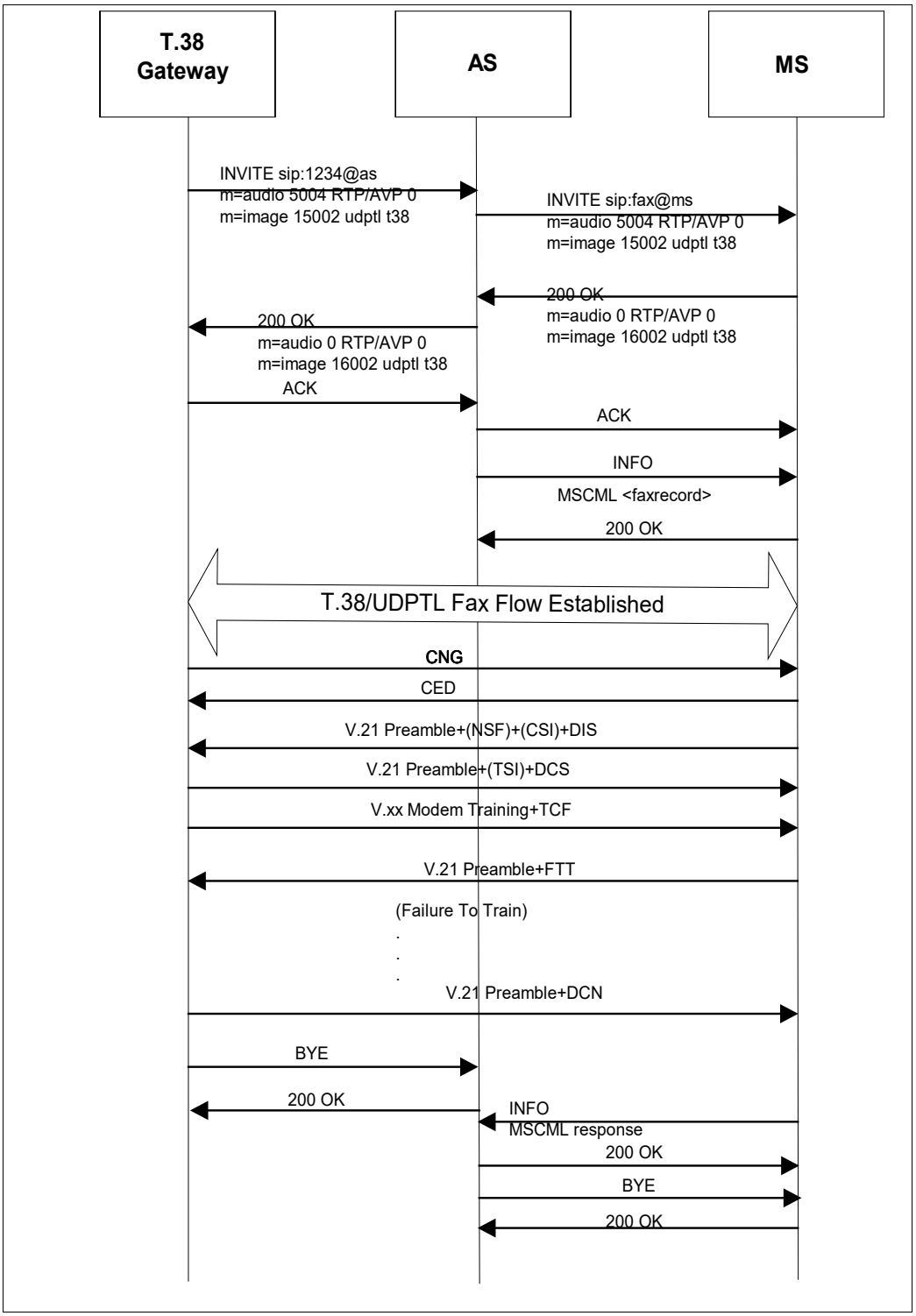

Figure 46 Example of Unsuccessful “faxrecord” Session

Once the image data is received, the Media Server stores it in a TIFF file and uses HTTP PUT to send the file to the URL provided by the *recurl* attribute in the MSCML <faxrecord> request. The screen shot of the "faxrecorded" TIFF file is shown in *Figure 47*. Note that this document can have up to two fax headers inserted respectively by the fax machine and the Media Server. (If the fax machine does not insert its own header, there is only one fax header inserted by the Media Server). This also means that up to three fax headers may appear in the fax when the Media Server does "faxplay" of the previously recorded TIFF file. (There are two fax headers in the printed fax document, if the document was recorded from the fax machine that does not insert its own header).

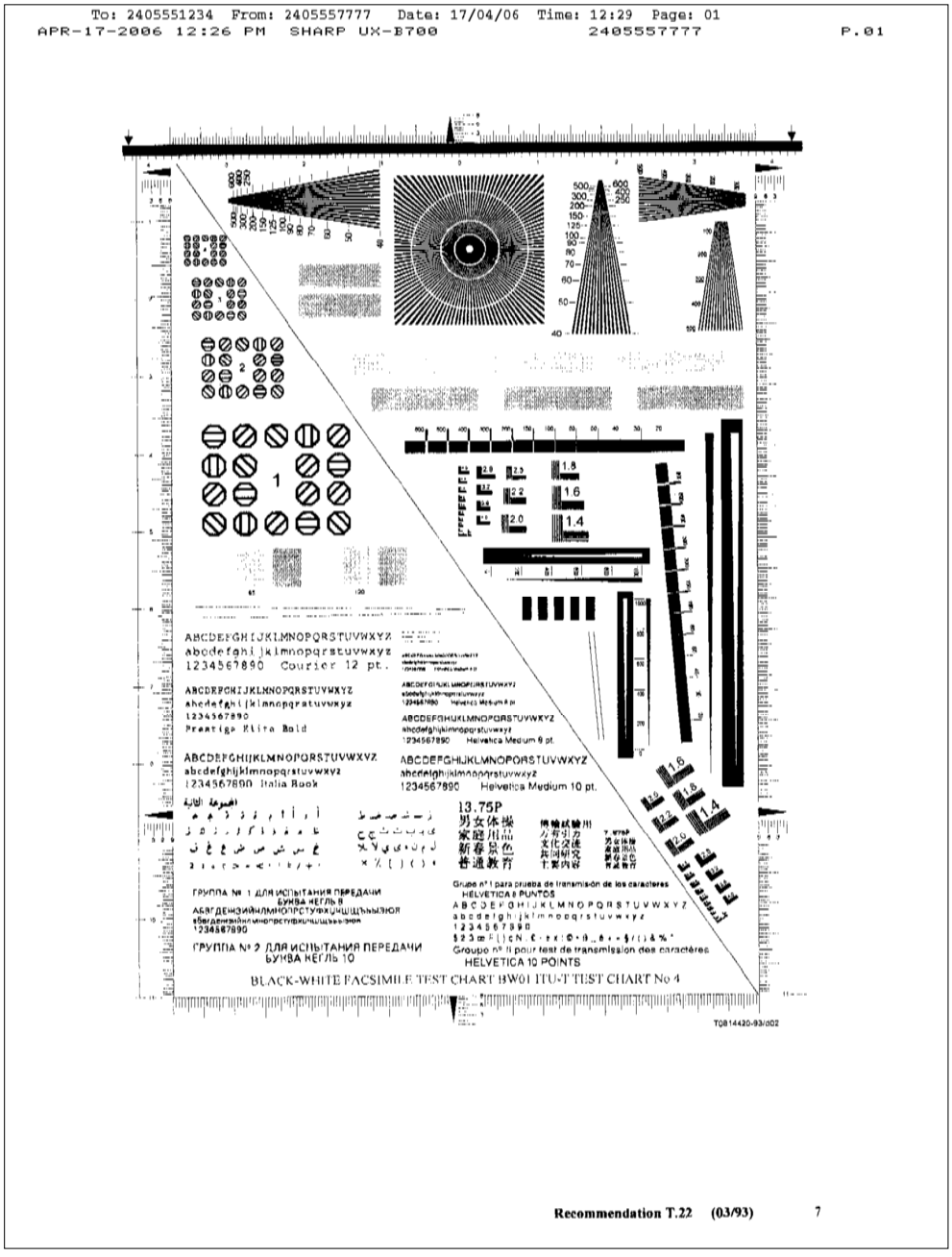

Figure 47 Screen Shot of One Page "faxrecorded" TIFF File with Two Fax Headers

3.28.2 Fax Printing

Cisco BroadWorks sends outgoing fax calls via the Cisco BroadWorks voice portal telephone user interface. The outgoing call to the remote fax machine is established with only an audio codec to allow the local subscriber to hear any signals or treatments that may be encountered while the call is proceeding. Once the remote fax machine answers, the call is dynamically switched to a fax call, as described in T.38 section D.2.2.4.

The “Send” mode of the T.30 protocol is supported. The “Poll” and “Turnaround Poll” modes are not supported.

3.28.2.1 Message Sequence Diagrams

A successful “faxplay” session setup is shown in *Figure 48*. Note that the HTTP GET operation, which occurs right after the reception of the INFO <faxplay> request message from the Application Server, is omitted from the following call flows.

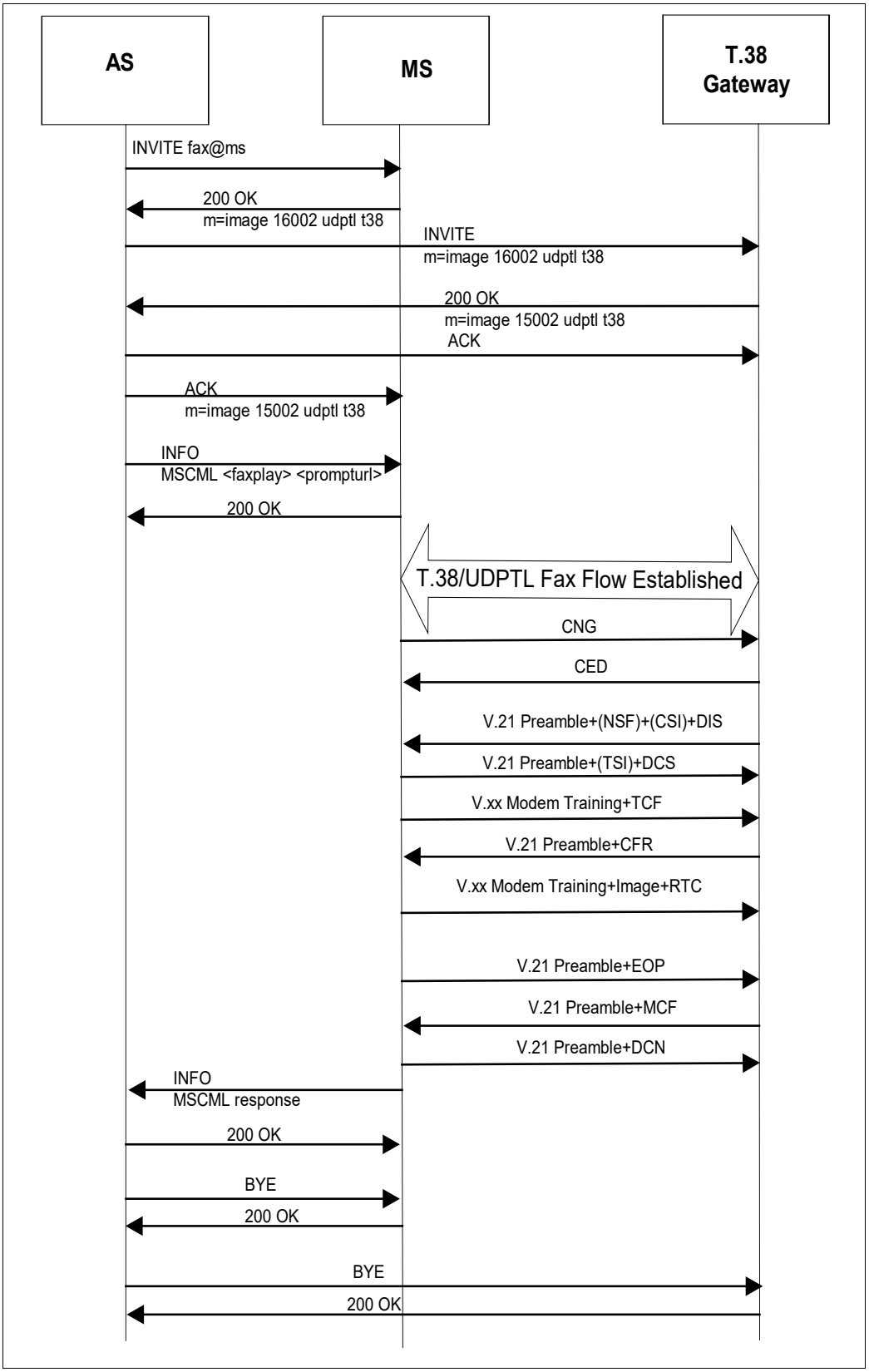

Figure 48 Example of Successful "faxplay" Session

A successful “faxplay” session setup (re-INVITE from Terminating T.38 Gateway scenario) is shown in *Figure 49*.

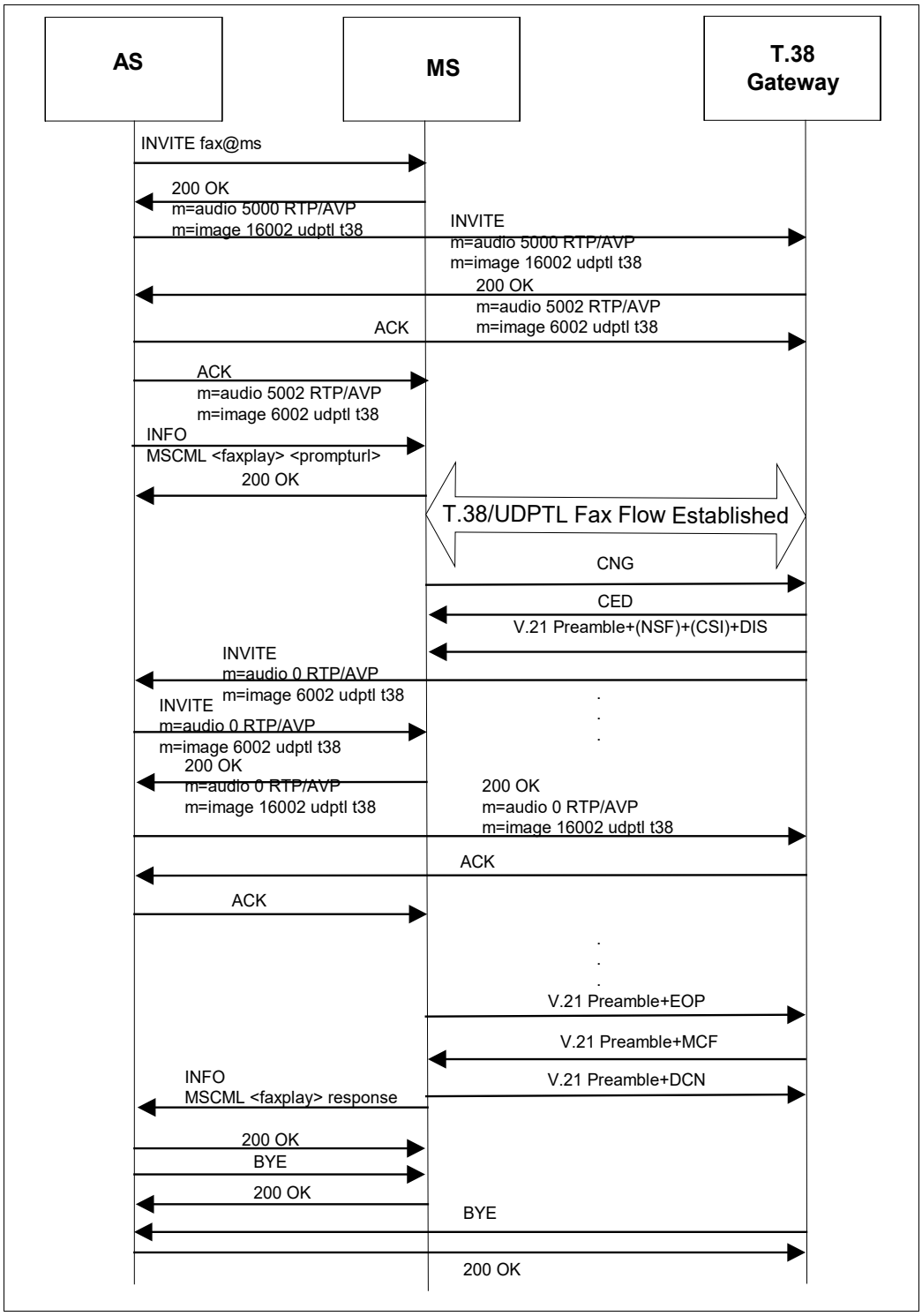

Figure 49 Successful “faxplay” Session in Re-INVITE from Terminating T.38 Gateway Scenario

A successful “faxplay” session setup (re-INVITE from Originating T.38 Gateway/Media Server scenario) is shown *Figure 50*.

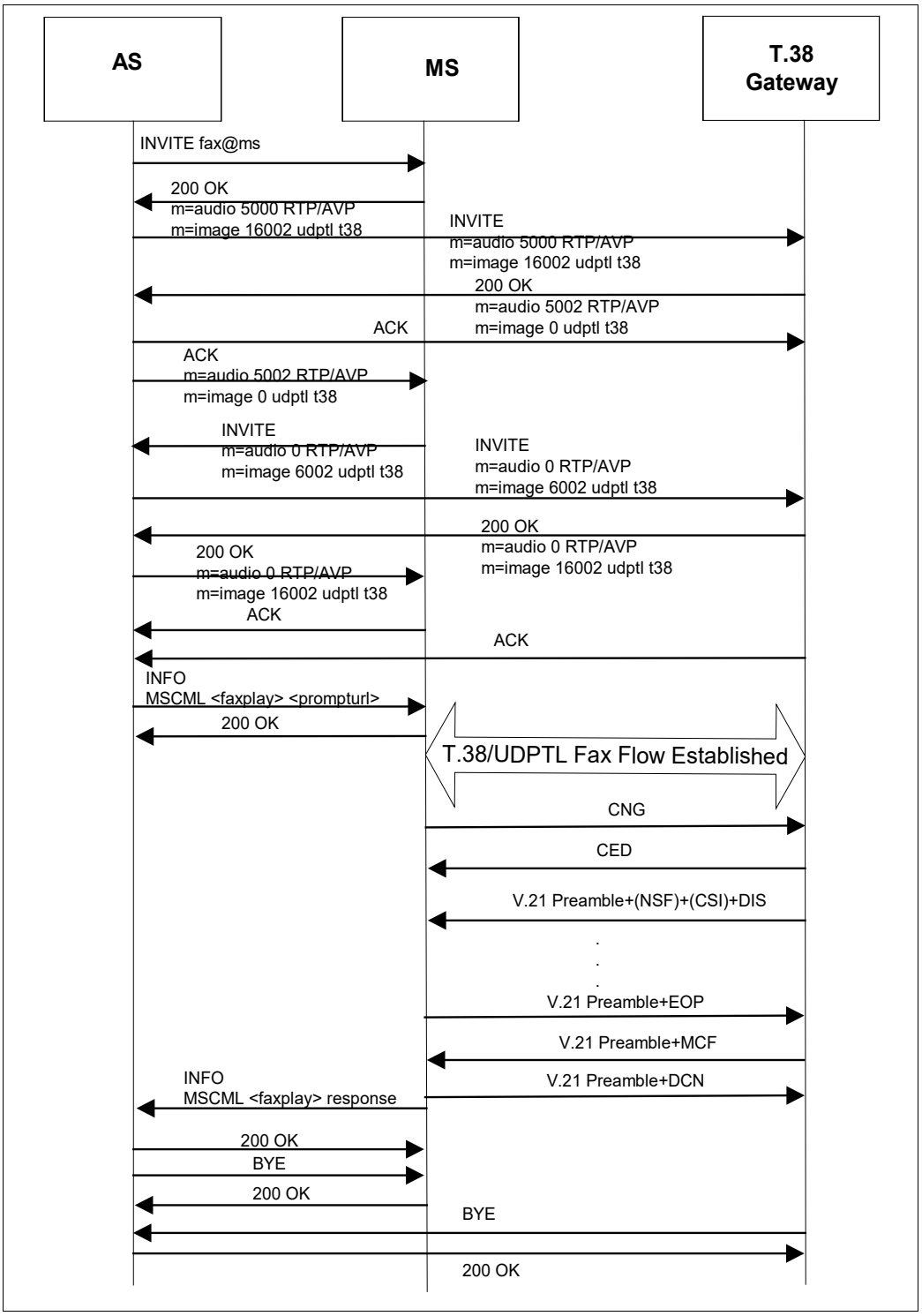

Figure 50 Successful “faxplay” Session in Re-INVITE from Originating T.38 Gateway (Media Server) Scenario

Figure 51 shows an unsuccessful “faxplay” session setup.

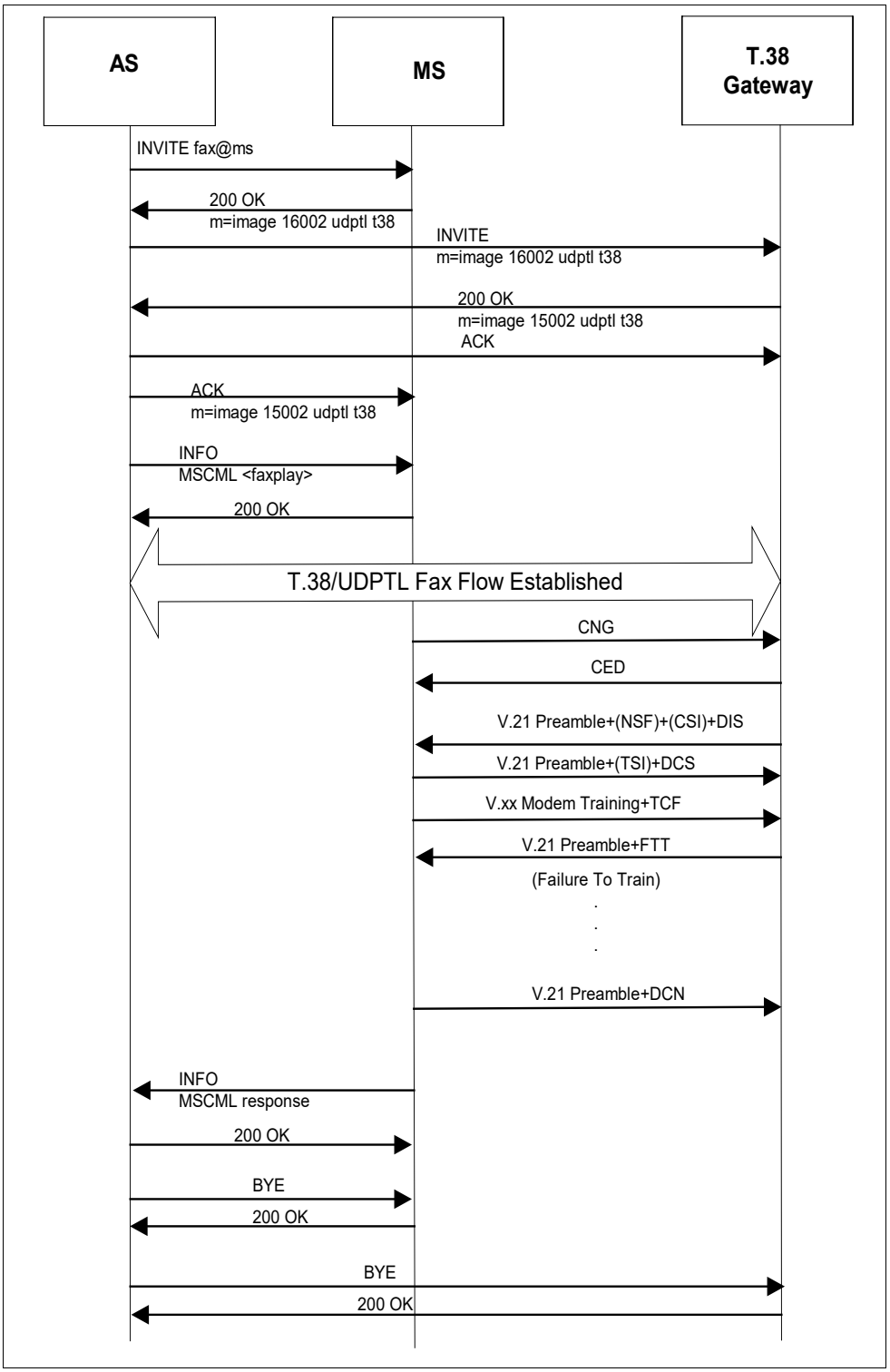

Figure 51 Example of Unsuccessful “faxplay” Session

3.29 SDP: Session Description Protocol (RFC 4566)/Support for IPv6 in Session Description Protocol (RFC 3266)/Offer/Answer Model with Session Description Protocol (RFC 3264)

The Session Description Protocol (SDP) is used to specify media characteristics for a session. A network device must support certain types of session descriptions for Cisco BroadWorks to provide enhanced call control services, such as Call Hold. Specifically, a device must support SDP usage, as specified in *RFC 3264*. Cisco BroadWorks is fully compliant with *RFC 4566* and *RFC 3264*. As IPv6 networks become more prevalent, devices should support *RFC 3266 - Support for IPv6 in Session Description Protocol (SDP)*. Cisco BroadWorks will support *RFC 3266* in future releases.

Cisco BroadWorks "brands" all SDPs that pass through Cisco BroadWorks such that the devices exchanging SDPs view Cisco BroadWorks as the owner of the SDP. Cisco BroadWorks changes the **v, o,** and **s** lines as part of branding an SDP. For example, a device initiates a request to Cisco BroadWorks with a device-generated SDP. Cisco BroadWorks overwrites the **v**, **o**, and **s** lines of the SDP from the device, leaving the media descriptions intact. Cisco BroadWorks then sends the modified SDP to the terminating device. Cisco BroadWorks manipulates the SDP to ensure SDP interoperability between devices. Specifically, Cisco BroadWorks ensures that the devices receive valid SDPs adhering to the rules of *RFC 3264* and *RFC 4566*, especially for the session ID and version contained in the **o** line.

Cisco BroadWorks also can interwork devices that use the *RFC 3264* hold mechanism with devices that do not yet support the *RFC 3264* hold mechanism. Cisco BroadWorks converts an SDP with an *a* attribute to sendonly or inactive with an SDP, which has a connection address (c line) of 0.0.0.0 when sending the SDP to a device that does not support the *RFC 3264* hold mechanism. Additionally, Cisco BroadWorks always appends or corrects the appropriate *a* attribute in a received SDP according to the *a* attribute media direction rules defined in *RFC 3264*. For example, if Cisco BroadWorks receives an SDP with a **c** line of 0.0.0.0, with no *a* attribute, Cisco BroadWorks modifies the SDP to include an *a* attribute of inactive. Note for sendrecv SDPs, Cisco BroadWorks does not include the *a* attribute, as this is the default mode for a SDP.

Cisco BroadWorks also fully supports SDPs with multiple media streams ("m=" lines) in an SDP. Cisco BroadWorks is able to parse and act on multiple **m** lines within an SDP. Additionally, Cisco BroadWorks considers an SDP as "held" from a service call processing perspective, if all active audio media streams are held. An active media description is one where the port field in an **m** line is not 0. For example, if a device in active call with another device provides to Cisco BroadWorks an SDP with multiple media streams, and each stream is held, Cisco BroadWorks considers this SDP as a "held" SDP and applies the appropriate "hold" service to the other device. For example, an appropriate "hold" service could be Music On Hold.

A network device must support the following actions:

- A network device must support receiving a subsequent INVITE (Re-INVITE) with a new session description. The device should swap media streams transparently.
- A network device must support receiving an initial INVITE with a session description where the connection addresses ("c=" lines) for the media streams are set to zero (0.0.0.0). (Note that this method of putting a device on hold is deprecated, but Cisco BroadWorks supports it for backwards compatibility.) A network device should support receiving an initial INVITE with a session description, where the *a=inactive* attribute is present to indicate a particular media stream is inactive (that is, on hold). The network device should make use of the *a=sendonly*, *a=recvonly*, *a=inactive*, *a=sendrecv* (default) attributes in the session description it provides, to indicate the disposition of the media it is providing.

- A network device must support receiving a subsequent INVITE request (re-INVITE) with a modified session description where the connection addresses ("c=" lines) for the media streams are set to zero (0.0.0.0). (Note that this method of putting a device on hold is deprecated, but the network device should support it for backwards compatibility.) A network device should support receiving a subsequent INVITE request (re-INVITE) with a modified session description, where the a=inactive attribute is present to indicate a particular media stream is inactive (that is, on hold). The network device should make use of the *a=sendonly*, *a=recvonly*, *a=inactive*, *a=sendrecv* (default) attribute in the session description it provides, to indicate the disposition of the media it is providing.
- A network device should support receiving an initial INVITE request with no session description. In this scenario, the device should be able to handle setting up a media connection when it receives an ACK request with media. Additionally, the network device should respond with an SDP in the 200 (OK) response to the INVITE request, which contains all of the supported codecs of the device in the order of preference. A network device should also support receiving an initial INVITE request with a session description that contains no media lines ("m=" lines). A network device should respond with a session description with no media lines in the 200 (OK) response to the INVITE request.
- A network device must support receiving a subsequent INVITE request (re-INVITE) with no session description. In this scenario, the device should be able to handle setting up a media connection when it receives an ACK request with media. Additionally, the network device should respond with a session description in the 200 (OK) response to the INVITE request, which contains all of the supported codecs of the device in the order of preference.
- A network device must support receiving an ACK to an INVITE, which did not have a Session Description, with a Session Description where the connection addresses ("c=" lines) for the media streams are set to zero (0.0.0.0). Note that this method of putting a device on hold is deprecated, but must be supported for backwards compatibility. A network device should support receiving an ACK to an INVITE, which did not have a Session Description, with a Session Description where the *a=inactive* attribute is present to indicate a particular media stream is inactive (that is, on hold). The network device should make use of the *a=sendonly*, *a=recvonly*, *a=inactive*, *a=sendrecv* (default) attribute in the Session Description it provides, to indicate the disposition of the media it is providing.
- Upon receiving a subsequent INVITE (Re-INVITE) with no Session Description, a network device should return a Session Description that results in connecting the media streams. A device should not return a *200 OK* response with a Session Description where the connection addresses ("c=" lines) for the media streams are set to (0.0.0.0) or containing an a=inactive, unless it intends to place the call "on-hold". Note that this method of putting a device on hold (c=0.0.0.0) is deprecated, but must be supported for backwards compatibility.
  - This scenario can occur when the device is placed "on-hold" via a Session Description, where the connection addresses ("c=" lines) for the media streams are set to zero (0.0.0.0) or contain an a=inactive attribute and receive a subsequent INVITE (Re-INVITE) with no Session Description (Null SDP).
  - The intention of the subsequent INVITE (Re-INVITE) is to re-establish the media path(s). Therefore, a device should not return a *200 OK* response with a Session Description, where the connection addresses ("c=" lines) for the media streams are set to zero (0.0.0.0) or contain an *a=inactive* attribute, unless it intends to keep the call "on-hold".

- A network device must support changing media streams in 18x responses and subsequent *200 OK* responses. For calls from a network device to a Cisco BroadWorks user, it is noted that the SDP information sent in a *180 Ringing*, *183 Session Progress*, *200 OK* response, and subsequent INVITEs (Re-INVITEs) can all contain different media descriptions. Cisco BroadWorks may forward a call from a user to voice mail or transfer from the Auto Attendant to a user, which in both cases alters the media stream.
- Upon receipt of an 18x response with media, the network device should stop providing local ringback and rely on the remote side for its "progress announcements". This could occur in a no-answer forward scenario or transfer before answer scenario. The network device does not have to support switching from remote ringback to local ringback, as Cisco BroadWorks does not initiate this transition. However, it is desirable for the network device to support switch from remote to local ringback.

For more information on SDP, see *Appendix A: SDP Overview*.

3.29.1 Cisco BroadWorks Content-type Support

Cisco BroadWorks supports session establishment for requests with `application/sdp` content-types. In addition, Cisco BroadWorks also supports session establishment of other content-types via system configuration. The `application/sdp` is the only content-type allowed by the Application Server by default. The Cisco BroadWorks Application Server rejects unrecognized content-types not explicitly provisioned on the Application Server with a 415 Unsupported Media Type response.

Requests with bodies, with `multipart/mixed` types are also implicitly supported, requiring no configuration. The following rules summarize Cisco BroadWorks content-type handling in multipart/mixed bodies:

- All content types recognized by Cisco BroadWorks (that is, explicitly configured) are allowed and proxied by Cisco BroadWorks. Cisco BroadWorks maintains the multipart/mixed body when more than one content-type is recognized by Cisco BroadWorks.
- All content types not recognized by Cisco BroadWorks are silently discarded when contained within a multipart/mixed body.
- If no content types are recognized by Cisco BroadWorks in a multipart/mixed body, Cisco BroadWorks rejects the request with a *415 Unsupported Media Type* response.
- If only one content type is recognized by Cisco BroadWorks, the multipart/mixed body is dropped and replaced with a normal body containing the single content type.

All content types added to the supported content-type list via the Application Server CLI are proxied by Cisco BroadWorks when received. Cisco BroadWorks performs content-sensitive processing to the following content types: *application/sdp, application/gtd, application/broadsoft, application/dtmf-relay, audio/telephone-event,* and *application/dtmf*. Cisco BroadWorks does not perform any context-sensitive processing on other content types added to the supported content-type list via the Application Server CLI; these content types are simply proxied to the remote party.

Cisco BroadWorks does not currently use the *Content-Disposition* header for *Content-Type* header processing.

The following table shows how the Content-Type is managed for different SIP method and responses.

Method	Content-Type	Proxy to Access Devices and network	Proxy to Media Server
All requests and responses	Not configured under AS_CLI/Interface/SIP/ContentType	No	No
All requests and responses applicable to SDP	Application/sdp	Yes	Yes
INFO	Application/dtmf-relay	Yes	Yes
INFO	Application/dtmf and audio/telephone-event	Yes	Yes after conversion to application/dtmf -relay
INFO	Other	Yes	Yes
INVITE, PRACK, ACK, UPDATE, 2xx responses to INVITE, PRACK, UPDATE	All types excluding application/sdp	Yes	No
Other methods	All	No	No

Table 2 Content-Type Proxying Rules

For special rules applicable to the INFO method, see section *3.24 SIP INFO Method (RFC 2976, RFC 6086).*

3.30 RTP: Transport Protocol for Real-Time Applications (RFC 3550)/RTP: Transport Protocol for Real-Time Applications (RFC 1889)/RTP Profile for Audio and Video Conferences with Minimal Control (RFC 3551)/RTP Profile for Audio and Video Conferences with Minimal Control (RFC 1890)

Cisco BroadWorks uses this functionality within the Media Server to provide media resources for voice mail, IVR, conferencing, and so on.

Note that the Cisco BroadWorks Media Server supports the following encodings:

- G.711 u-law
- G.711 a-law
- G.726-32
- G.729a

Network devices must support codec renegotiation via SIP when using codecs other than G.711 u-law, G.711 a-law, G.726-32, and G.729a to interface with the Cisco BroadWorks Media Server. For calls using resources on a Media Server, Cisco BroadWorks must renegotiate the media stream to G.711 u-law, G.711 a-law, G.726-32, or G.729a.

Note that all devices that interface with Cisco BroadWorks MUST support G.711 u-law. This is required so that all of the devices on Cisco BroadWorks can communicate with each other. As an example, a Conferencing Server may only support G.711 u-law. Any devices that want to communicate with the Conferencing Server must offer G.711 u-law for a call to be set up with the Conferencing Server. The devices can and should place G.711 u-law at the end of the preference list so that G.711 u-law is only used when corresponding devices can only support G.711 u-law. All other devices would negotiate to the codec of choice according to the rules defined in *RFC 3264*.

3.31 Cisco BroadWorks Redundant Application Server Requirements

Cisco BroadWorks provides complete reliability via a geo-redundant network architecture. Cisco BroadWorks provides both network reliability and server reliability. Within this architecture, Cisco BroadWorks has multiple Application and Network Servers that may serve the network devices. To ensure that Cisco BroadWorks can service calls in a failure situation, certain requirements are imposed on the network devices.

The network device must be able to route to multiple addresses under failure conditions. This can be accomplished in many ways. Following are some of the possible mechanisms to support this requirement. A network device must support at least one of the possible mechanisms described.

- The network device should support a fully qualified domain name (FQDN) entry for the route to Cisco BroadWorks, as well as support DNS SRV records for resolution of the route address.

  The network device should attempt to route SRV processing rules on each entry, in the order specified by DNS, until a successful route is obtained. The network device should advance to the next route after receiving Internet control message protocol (ICMP) errors or timing out on the current route. Note that the timeout interval should be relatively short, to prevent longer than desirable call setup delays in a failure situation. The timeout interval should also be configurable.

- The network device should support an FQDN entry for the route to Cisco BroadWorks, as well as support DNS getAllHostsByName for resolution of route address A records.

  The network device should attempt to route on each A record in the order received by the DNS lookup, until a successful route is obtained. The network device should advance to the next A record after receiving ICMP errors or timing out on the current A record. Note that the timeout interval should be relatively short, to prevent longer than desirable call setup delays in a failure situation. The timeout interval should also be configurable.

- The network device should provide explicit support for a primary and secondary route to Cisco BroadWorks.

  The network device should use the secondary route to send messages, after receiving ICMP errors or timing out on the primary route. Note that the timeout interval should be relatively short, to prevent longer than desirable call setup delays in a failure situation. The timeout interval should also be configurable.

In addition to the above requirements, network devices should also support the following requirements.

- The network device should support a timer mechanism to route advance to the next address. This timer should take precedence over the SIP retransmission rules, to ensure timely routing of messages to Cisco BroadWorks in a failure situation.

- The network device should support the ability to send INVITEs and other requests to any of the addresses, as specified in the above requirements.
- The network device may also support state information as to the last successful route used to communicate with Cisco BroadWorks. In this scenario, the network device would remember the last successful route used to communicate the proxy for a duration of time. The network device would continue to use this "cached" route until the "cache" expires. If the network device supports this stateful capability, it must update the state when messages are received from Cisco BroadWorks from a different address than the current state. This update is required to allow Cisco BroadWorks to serve the subscriber from any of the Application Servers in the redundant cluster.
- The network device must support receiving a 3xx response with multiple contacts.
- The network device must support advancing through all the contacts, including all addresses that the contacts may resolve to until a successful route is obtained.
- The network device must stop contact advancement as a result of a 3xx response with multiple contacts, upon receipt of a global negative response (for example, *600 Busy Everywhere*).

  The network device, upon receipt of a 3xx response, should try all contacts in the 3xx response until a successful route is obtained. However, the network device should cease trying additional contacts upon receipt of a 6xx negative response. For example, the network device receives multiple contacts in a 3xx response. It tries the first contact that results in reaching the subscriber, but the subscriber is busy. The Cisco BroadWorks Application Server returns a *600 Busy Everywhere* to indicate that no other contacts should be attempted in trying to reach this subscriber. The network device should cease the contact advancement and signal the appropriate treatment for the originating end user.

3.32 Cisco BroadWorks Overload Handling Requirements

Cisco BroadWorks provides the ability to shed traffic in overload conditions of a particular Application Server. The Application Server actively detects overload conditions and redirects calls from devices to the secondary Application Server during periods of overload.

The Application Server can be configured to take one of the following actions for calls that are received during overload conditions:

- Ignore.
- Return a 302 Moved Temporarily response, to redirect the call to the other Application Server.
- Return a 503 Service Unavailable response to redirect the call to the next available address in the device's routing list.

When the Application Server is configured to ignore, the Application Server ignores requests for new calls and the device making the call must attempt the secondary server, using the procedures described in section *3.31 Cisco BroadWorks Redundant Application Server Requirements*.

When the Application Server is configured to return a 302 Moved Temporarily response, the Application Server returns a 302 Moved Temporarily response containing a contact with a *maddr* parameter containing the address of the other Application Server, for the following requests when received outside an existing dialog: INVITE, BYE, OPTIONS, NOTIFY, SUBSCRIBE, REGISTER.

For example, if the following INVITE is received on the Application Server:

```
INVITE sip:3013330000@ascluster1.broadsoft.com SIP/2.0
```

… then the Application Server returns a *Contact* header with the address of the other Application Server in the *maddr* parameter as shown in the following example.

```
Contact:<sip:3013330000@ascluster1.broadsoft.com:5060;maddr=
as2.broadsoft.com>
```

The device must honor the received contact in the 302 response and send a subsequent INVITE to the address specified in the *maddr* parameter, populating the *Request-URI* of the INVITE with the SIP-URI contained in the contact received in the 302 response.

When the Application Server is configured to return a *503 Service Unavailable* response, the expected behavior by the source of the message is defined in *RFC 3261*, section *21.5.4*. The device must attempt the next route obtained from the DNS SRV lookup, as specified in *RFC 3261*, section 28.1, and in *RFC 3263*, section 4.3. By following these RFC procedures, the device effectively contacts the other Application Server during overload conditions.

When the Application Server is overloaded and returns a *503 Service Unavailable* response, it can include a *Retry-After* header field if the neighbor is configured to be a Retry-After receiver. The service unavailable period (Retry-After value) is a randomly chosen value within the provisioned range. If the Application Server receives a retransmission of the request that the Application Server has responded with Retry-After included in 503, the Application Server does not include the Retry-After in the 503 response for the retransmitted request.

When a neighbor is overloaded, it can reject a request from the Application Server using a 503 response with *Retry-After* header field. Upon receiving such a response, the Application Server considers the neighbor as overloaded. A time stamp indicating the service unavailable time period is set. During this service unavailable time period, the Application Server throttles the SIP requests sent to the overloaded neighbor.

3.33 Cisco BroadWorks Video Device Requirements

BroadWorks provides support for video devices. BroadWorks has specific requirements related to the video processing for devices.

3.33.1 Cisco BroadWorks Video Add-On Support

Cisco BroadWorks supports both integrated video devices and video-only devices for the Cisco BroadWorks Video Add-On service. The Cisco BroadWorks Video Add-On service allows an additional video-capable device to be configured on a subscriber if the subscriber's primary device does not support video, while allowing the subscriber to use their primary device for audio.

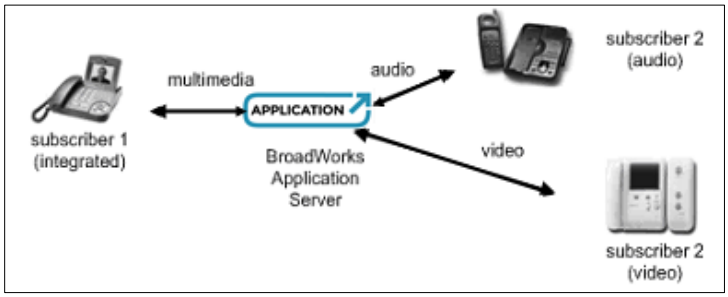

Figure 52 Cisco BroadWorks Video Add-On Service

- **Integrated video device**: An integrated video device has multimedia capability, including the ability to have both audio and media streams. The signaling flow of an integrated video device is identical to an audio-only device. The media for an integrated device includes both audio and video streams, providing both voice and video capabilities.
- **Video-only device**: A video-only device is a device used in conjunction with Cisco BroadWorks, which only uses video streams without any audio streams. Cisco BroadWorks "splits" incoming multimedia calls to the video-only device, directing the audio portions to the primary device of the Cisco BroadWorks subscriber and the video portions to the Video Add-On video-only device. Calls originated by the subscriber must occur from the audio device. Cisco BroadWorks invites the video-only device on originations. Additionally, Cisco BroadWorks blocks calls originated by the video-only device when used as part of the Cisco BroadWorks Video Add-on service.

Cisco BroadWorks has the following requirements for integrated video devices:

- An integrated video device must have the ability to negotiate separate audio and video media streams. This means that the device must be able to support multiple m lines per connection address of different media types, and the device should be able to support multiple connection addresses, each with one or more m lines.
- An integrated video device must have the ability to provide a multimedia offer, including both audio and video m lines in a *200 OK* response to an initial INVITE, or subsequent INVITE (Re-INVITE) with no Session Description (Null SDP).

Cisco BroadWorks has the following requirements for video-only devices used for the Cisco BroadWorks Video Add-On service:

- A video-only device must have the ability to provide a multimedia offer, including both audio and video m lines in a *200 OK* response to an initial INVITE or subsequent INVITE (Re-INVITE) with no Session Description (Null SDP).
- A video-only device must have the ability to negotiate a video-only media stream. This means that the device must be able to support a single m line of a video media type including:
  - Receiving a video-only offer containing a single m line of video media type.
  - Receiving a multimedia answer containing multiple m lines with an inactive audio m line via a media attribute and an active video m line.
- A video-only device must have the ability to accept an inactive video-only offer/answer, where the SDP contains a single m line of type video with a media attribute of inactive.
- A video-only device must have the ability to auto-answer an incoming call via either of the following mechanisms:
  - Support for the *Call-Info auto-answer* parameter.
  - Client configuration for auto-answer (this is a local policy on the device to auto-answer all incoming calls).

3.33.2 Cisco BroadWorks Video IVR Support

In addition to the Video Add-On service, Cisco BroadWorks also supports video IVR functions with the Media Server. The video-enabled Media Server provides the following functions:

- Play synchronized audio and video streams over RTP.
- Record synchronized audio and video streams over RTP.
- Play synchronized audio and video streams over RTP, while performing dual-tone multi-frequency (DTMF) digit collection simultaneously.
- Play the audio portion of an audio + video file.
- Record a synchronized audio and video stream, while a separate audio and video stream are played.

The Media Server supports the video codec H.263 along with the MOV file format for storing audio and video. The WAV file format is still supported for audio-only RTP streams.

3.33.2.1 Video File Format

The Media Server supports the “Hinted” .MOV file format for playing and recording video. The Media Server has the capability to play back and record movie files (.mov) encoded using the H.264 video codec. The WAV file format is still supported for audio-only RTP streams.

The Apple QuickTime media player is required when playing a movie back that was e-mailed to a subscriber by the Media Server on a PC. A common service that sends such e-mails is Visual Voice Mail.

The Apple QuickTime Pro media player is required when recording on a PC a movie that will be uploaded to the Application Server for later play back by the Media Server. Note that Apple QuickTime Pro does not support transcoding to H.263-1996, which is the only H.263 video codec supported by Windows Messenger 5.0/5.1. This means that the Media Server cannot receive or stream video to Windows Messenger.

3.33.2.2 Video Codecs Supported

The Cisco BroadWorks Media Server supports the H.263-1998, H.263-2000, and H.264 video codecs. The Media Server does not perform any video transcoding. If a video device does not support the video format in which a file was recorded, the video is not displayed.

Note that H.263-1998 and H.263-2000 are compatible with each other (only SDP parameters change between these two H.263 variants). Therefore, the Media Server is able to play H.263-1998 files to an H.263-2000 video device, and play most H.263-2000 files to an H.263-1998 video device.

For the H.263-1998 video codec, all annexes must be disabled, which makes H.263-1998 backward compatible with H.263-1996.

For the H.263-2000 video codec, the Media Server supports H.263 Annex X profile 0, and levels 10 and 20. Profile 0 is compatible with H.263-1996. Level 10 represents a resolution of 176 x 144 pixels at 15 frames/second and a maximum bit stream of 64 KB/sec. Level 20 represents a resolution up to 352 x 288 pixels at 30 frames/second and a maximum bit stream of 128 KB/sec.

3.33.2.3 Cisco BroadWorks Video Device Requirements for Video IVR

Cisco BroadWorks has the following requirements for video devices utilizing the Cisco BroadWorks Video IVR functionality:

- Support of *draft-ietf-avt-rfc2429-bis-04*
- Support of *draft-levin-mmusic-xml-media-control-03* (fast-update primitive only)
- Support of the 1996 edition of H.263 using the *RFC 2949* RTP payload. This is also known as:
  - H.263-2000 Annex X profile 0, or
  - H.263-1998 with all annexes disabled

3.33.2.4 SDP Handling – Video Streaming Enabled on Media Server

When the Media Server receives an offer SDP and video streaming is enabled, it looks for the presence of a video line. If one is found, the following logic is performed:

For H263-1998 RTP payloads if H263-1998 is enabled:

- If the custom picture size format option is received, the Media Server recognizes whether it indicates a picture size smaller than quarter common intermediate format (QCIF) or a picture size larger than common intermediate format (CIF). Otherwise, it ignores the custom format option even if the dimensions in the custom format option correspond to CIF or QCIF.
- If the largest recognized picture size format option indicates a picture size smaller than QCIF, the RTP payload is rejected.
- Otherwise, MPI values for the CIF and QCIF format options are stored. If any of these options are missing and a larger picture size was provided in the offer SDP, then default values are stored as follows in the order presented:
  - If CIF is missing and a picture size larger than CIF was provided in the offer SDP, CIF is defaulted to “2”.
  - If QCIF is missing and a picture size larger than QCIF was provided in the offer SDP, QCIF is defaulted to that of CIF divided by “2” (but with minimum result of “1”).
  - If the RTP payload is accepted, the corresponding payload in the answer SDP contains the following format options:
    - CIF with MPI determined as previously described
    - QCIF with MPI determined as previously described

For H263-2000 RTP payloads if H263-2000 is enabled:

- If profile or level is present, the profile must be “0” (baseline profile). All other format options (both H263-1998 format options and INTERLACE) are ignored. A profile indicating anything other than the baseline profile results in rejection of the RTP payload.
- If profile or level is not present, and there are no H263-1998 picture size or annex format options, this implies the baseline profile. The INTERLACE format option is ignored. The RTP payload is accepted.
- If the H264 RTP payload is accepted, the following values are stored and included in the answer SDP:
  - profile=0

- level=20

- If profile/level is not present, and there are H263-1998 picture size or annex format options, processing occurs according to the description for H263-1998 RTP payloads above.

For H264 RTP payloads if H264 is enabled:

- Packetization-mode must be present and must be "1" (non-interleaved). (*RFC 3984* states that packetization-mode defaults to "0" if it's not present) Otherwise, the RTP payload is rejected.
- The profile component of profile-level-id, if present, must be "0" (baseline profile). (*RFC 3984* states that the profile defaults to "0" if profile-level-id is not present). A profile component indicating anything other than baseline profile results in rejection of the RTP payload.
- The Media Server supports the max-fs and max-mbps H.264 SDP payload format options. An endpoint uses these options to indicate that it can handle a greater resolution and frame rate than that supported by the H.264 level. The max-fs option specifies the maximum frame size in macroblocks (a macroblock is a 16 x 16 area). The max-mbps option specifies the maximum number of macroblocks the endpoint can decode per second. These options are specified in *RFC 3984*, *RTP Payload Format for H.264 Video*.
- The remaining format options (max-cpb, max-dpb, and max-br, redundant-pic-cap, parameter-add, sprop-interleaving-depth, sprop-deint-buf-req, deint-buf-cap, sprop-init-buf-time, sprop-max-don-diff, max-rcmd-nalu-size, and so on) are ignored.
- If the H264 RTP payload is accepted, the following values are stored and included in the answer SDP:
  - packetization-mode=1
  - profile-level-id with profile component "0", constraint flags component equal to that received in the offer SDP (or 0x00 if profile-level-id was not present in offer SDP), and level component equal to that received in the offer SDP (or 0x0A if profile-level-id was not present in offer SDP).

The following example shows an SDP with a valid video definition.

```
v=0
o=- 1038469523 1038469523 IN IP4 155.69.223.61
s=Optional title
c=IN IP4 155.69.223.61
t=0 0
m=audio 5002 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
m=video 5004 RTP/AVP 97 98
a=rtpmap:97 H263-1998/90000
a=rtpmap:98 H263-2000/90000
a=fmtp:97 CIF=2;QCIF=1
a=fmtp:98 profile=0;level=20
```

If the Media Server finds such a video line, it sends an answer SDP to the remote party.

```
v=0
o=BroadWks 758 0 IN IP4 192.168.4.178
s=Media Server SDP
c=IN IP4 192.168.4.178
t=0 0
m=audio 10992 RTP/AVP 0
```

```
m=video 10994 RTP/AVP 97
a=rtpmap:97 H263-1998/90000
a=fmtp:97 CIF=2;QCIF=1
```

The video codec that is negotiated (H263-1998 in the previous example) depends on parameter *MS_CLI/Services/IVR/IVRCodecList*.

If the Media Server finds a video line that contains only non-supported video codecs in the offer SDP (for example, H.261), the Media Server replies with an answer SDP with the video port deleted (that is, zeroed-out) to inform the remote party to disable the video stream.

```
v=0
o=BroadWks 758 0 IN IP4 192.168.4.178
s=Media Server SDP
c=IN IP4 192.168.4.178
t=0 0
m=audio 10992 RTP/AVP 0
m=video 0 RTP/AVP 31
a=rtpmap:31 H261/90000
```

The Media Server generates an offer SDP with a video line when it receives a SIP INVITE without an SDP and video streaming is enabled. An offer SDP sent by the Media Server contains the following format options if the corresponding codec(s) is enabled:

- For H263-2000 RTP payloads: profile=0;level=20
- For H263-1998 RTP payloads: cif=2;qcif=1
- For H264 RTP payloads: profile-level-id=0x42000C

```
v=0
o=BroadWks 758 0 IN IP4 192.168.4.178
s=Media Server SDP
c=IN IP4 192.168.4.178
t=0 0
m=audio 10992 RTP/AVP 0
m=video 10994 RTP/AVP 98
a=rtpmap:98 H263-1998/90000
a=fmtp:98 CIF=2;QCIF=1
```

When the Media Server receives the corresponding answer SDP, it performs the following logic:

For H263-1998 RTP payloads

The behavior, as described above, for H263-1998 RTP payloads received in an offer SDP applies, except that no SDP is sent in response.

For H263-2000 RTP payloads

The behavior as described above for H263-2000 RTP payloads in an offer SDP applies, with the following exceptions:

- If profile/level is present with the profile indicating “0” (baseline), the level received is the level stored (that is, the Media Server does not change it to “20”.)
- No SDP is sent in response.

For H264 RTP payloads

The behavior as described above for H264 RTP payloads in an offer SDP applies, except that no SDP is sent in response.

During video file playback, the Media Server compares the characteristics of the video file against the negotiated SDP format options to determine whether the video stream is compatible (that is can be played back).

An H.264 video file is considered compatible if

The level component of the sequence parameter set from the file does not exceed the level component of the profile-level-idc from the negotiated SDP format options. According to *ITU-T H.264 A.3.1*, special handling is required for level value "11". If the level is coded as "11" and the *constraint_set3_flag* is set, then the level shall be treated as "1b", which is higher than "1" (coded as "10") and lower than "1.1" (coded as "11").

If the negotiated SDP format options for H263 use profile/level, the H.263 video file is considered compatible if the file's picture size and video frame rate are within the limits of the negotiated level. The limits are based on table X.2 in *ITU-T H.263*:

- Levels 10 and 45 support a maximum picture size of QCIF with an MPI of 2 or higher.
- Level 20 supports CIF with an MPI of 2 or higher QCIF and smaller with MPI of 1 or higher.
- Levels 30 and 40 support CIF with an MPI of 1 or higher.
- Levels 50 and 60, when you eliminate picture sizes greater than CIF, support CIF at fps <= 50 and 352 x 240 at fps <= (60000/1001).
- Level 70 supports picture sizes >= CIF and frame rates >= 50 fps. The Media Server enforces a picture size <= CIF since the Cisco BroadWorks does not support picture sizes larger than CIF.

If the SDP format options stored against the codec do not have a profile/level, the H.263 video file is considered compatible if the picture size MPI for the file >= negotiated MPI of the corresponding picture size SDP format option.

3.33.2.5 SDP Handling – Video Streaming Disabled on Media Server

When video streaming is disabled on the Media Server and it receives an offer SDP with a video line, the Media Server replies with an answer SDP, with the video port deleted (that is, zeroed-out) to inform the remote party to disable the video stream.

```
v=0
o=BroadWks 758 0 IN IP4 192.168.4.178
s=Media Server SDP
c=IN IP4 192.168.4.178
t=0 0
m=audio 10992 RTP/AVP 0
m=video 0 RTP/AVP 97
a=rtpmap:97 H263-1998/90000
```

Also, when the Media Server generates an offer SDP and media streaming is disabled, the offer SDP does not contain a video line.

```
v=0
o=BroadWks 758 0 IN IP4 192.168.4.178
s=Media Server SDP
c=IN IP4 192.168.4.178
t=0 0
m=audio 10992 RTP/AVP 0
```

3.33.2.6 SDP Handling – Optional H.263 Parameters

A few expired internet-drafts (for example, *draft-even-avt-h263-h261-options-00.txt*) define options to control the bandwidth used by the video stream. These options appear in the SDP in an "a=" line. The Media Server will ignore these options until industry consensus is established in that area.

3.33.2.7 H.264 Parameter Sets Over RTP

The H.264 parameter sets carry information such as picture size and other video characteristics that apply for more than one frame. However, the information carried by H.264 parameter sets is not necessarily common among video files.

RFC 3984 specifies that H.264 parameter sets can be carried in both Session Initiation Protocol (SIP)/Session Description Protocol (SDP) offer/answer signaling and/or over the Real-Time Transport Protocol (RTP) stream.

The Media Server exchanges H.264 parameter sets over RTP. It does not exchange H.264 parameter sets over SIP signaling.

The Media Server does not issue a SIP re-INVITE if there is a change in H.264 parameter sets when asked to play more than one video file. However, the Media Server continues to re-invite the remote party when there is a codec modification between the playing media files.

3.34 Cisco BroadWorks 3GPP IMS Support/Private Header (P-Header) Extensions to SIP for 3GPP (RFC 3455)

Cisco BroadWorks supports the Third-Generation Partnership Project (3GPP) IMS architecture. In the 3GPP IMS architecture, the Cisco BroadWorks Application Server (AS) resides in the application layer and is a 3GPP-compliant Application Server. The Cisco BroadWorks Application Server interworks with a Call Session Control Function (CSCF). In particular, Cisco BroadWorks interfaces with the Serving CSCF (S-CSCF).

Cisco BroadWorks uses the *3GPP* headers defined in *RFC 3455* to interwork with 3GPP compliant S-CSCFs. Specifically, the Cisco BroadWorks Application Server uses and supports the following headers defined in *RFC 3455*:

- *P-Called-Party-ID* header
- *P-Charging-Vector* header

For more information about Cisco BroadWorks 3GPP IMS support, see the *Cisco BroadWorks AS Mode ISC Interface Specification* [40].

3.35 Cisco BroadWorks P-Early-Media Header Support (RFC 5009)

Reference Documents:

- *RFC 5009: Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media*, September 2007

3.35.1 Support for the P-Early-Media Header

The *P-Early-Media* header provides signaling information that allows a service provider to control early media flows from or to the terminating endpoint. Conceptually, a Gating Function allows or blocks early media, while a Policy Function implements the policies that determine whether early media should be allowed. (See the following figure.) The Gating Function may also provide ringback tone to the originating endpoint in some cases. The *P-Early-Media* header provides a way for the Policy Function to communicate the policy decision to the Gating Function, which actually implements the early media access control.

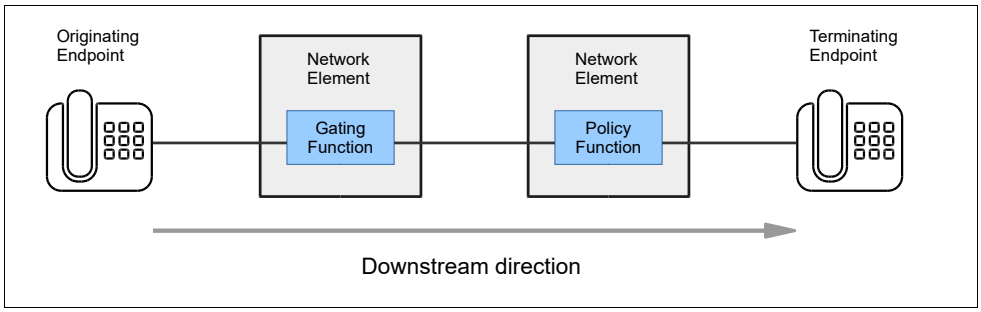

Figure 53 Gating Function and Policy Function

In most scenarios, Cisco BroadWorks does not play the role of the Gating Function or the Policy Function, but depends on external network elements to play these roles. However, in some scenarios Cisco BroadWorks may provide ringback, which a Gating Function might otherwise provide. In other scenarios, it may take the Policy Function role, particularly when Cisco BroadWorks manages early media streams from secondary endpoints (such as Shared Call Appearance endpoints) or when Cisco BroadWorks itself provides early media (such as via the Custom Ringback service).

Cisco BroadWorks' support for the *P-Early-Media* header is enabled when the SIP system parameter *supportPEarlyMediaHeader* is set to "true" and disabled otherwise. When PEM support is enabled, Cisco BroadWorks generally relays the *P-Early-Media* header in SIP messages before answer. Conversely, when PEM support is disabled, Cisco BroadWorks removes the *P-Early-Media* header from SIP messages. The remainder of this section describes Cisco BroadWorks behavior when PEM support is enabled.

When Cisco BroadWorks receives an initial INVITE request that contains a *P-Early-Media* header, it assumes the presence of an upstream network element that provides the Gating Function. *RFC 5009* states that the *P-Early-Media* header in the INVITE request should have the "supported" parameter. However, Cisco BroadWorks does not require this parameter and interprets[3] any *P-Early-Media* header as an indication that an upstream network element provides the Gating Function. Conversely, if the initial INVITE request does not contain a *P-Early-Media* header, then Cisco BroadWorks assumes that the Gating Function may be absent.

[3] The draft document that preceded RFC 5009, draft-ejzak-sipping-p-em-auth-02.txt, states that a *P-Early-Media* header in an INVITE request should contain no parameters. RFC 5009 added the "supported" parameter. BroadWorks supports both draft-ejzak-sipping-p-em-auth-02.txt and RFC 5009.

When Cisco BroadWorks sends an outgoing initial INVITE request, it adds a *P-Early-Media* header with “supported”.

For a basic call scenario involving a single terminating endpoint, Cisco BroadWorks relays the
P-Early-Media header in SIP messages before answer. These messages include:

- 18x provisional response to INVITE
- PRACK request
- 200 response to PRACK
- UPDATE request from originating endpoint or terminating endpoint
- 200 response to UPDATE

Cisco BroadWorks relays the *P-Early-Media* header in these SIP messages regardless of whether the initial INVITE request contained a *P-Early-Media* header.

Cisco BroadWorks supports a default PEM value, which it can apply to an incoming provisional response that omits a *P-Early-Media* header. Cisco BroadWorks applies the default PEM value to the first provisional response in the dialog that contains SDP (and omits the *P-Early-Media* header). This default value is configured as the SIP system parameter *defaultPEarlyMediaValue*, which can take the values “sendonly”, “inactive”, and “none”. When the parameter is set to “sendonly” or “inactive”, Cisco BroadWorks applies the PEM value “sendonly” or “inactive”, respectively, if the provisional response does not have a *P-Early-Media* header. The effect is the same as if the provisional response contained a *P-Early-Media* header with the configured value supplied for each media stream. If the parameter is set to “none”, then Cisco BroadWorks does not apply a default PEM value if the provisional response omits the *P-Early-Media* header.

When Cisco BroadWorks generates early media via the Media Server, it adds the *P-Early-Media* header to the provisional response along with the Media Server SDP. The following are some of the scenarios in which Cisco BroadWorks generates early media via the Media Server:

- Treatments (depending on the no charge option)
- Intercept (depending on the no charge option)
- Custom Ringback
- Call Waiting Ringback
- Sequential Ring comfort announcements
- Voice Mail (VM) Deposit warning tone

In most of these scenarios, Cisco BroadWorks sends *P-Early-Media* with the “sendonly” parameter for the Media Server media stream. However, if the Media Server needs to receive DTMF digits, then Cisco BroadWorks sends the “sendrecv” parameter instead.

Cisco BroadWorks may normalize a received PEM header before relaying it, so that it applies to the SDP as required by *RFC 5009* (one direction parameter per media stream), applying the following rules:

- If the received the *P-Early-Media* header has more direction parameters than the SDP has media streams, then Cisco BroadWorks removes the extra parameters from the *P-Early-Media* header.

- If the received the *P-Early-Media* header has fewer direction parameters than the SDP has media streams, then Cisco BroadWorks adds additional parameters as necessary. The added parameters have the same value as the last parameter in the received
P-Early-Media header.
- If Cisco BroadWorks normalizes an outgoing SDP by adding inactive m-line to it, it also adds a corresponding parameter in the *P-Early-Media* header with the value "inactive".

3.35.2 Interactions with Early Media Transitions

An early media transition occurs when a terminating endpoint begins sending early media (for example, remote ringback), then stops sending early media, requiring a transition to local ringback. If PEM support is enabled, Cisco BroadWorks can examine the *P-Early-Media* header to determine when an early media transition is needed. In a typical scenario, the terminating endpoint sends a reliable provisional response with SDP and "sendrecv" in the *P-Early-Media* header, indicating that it is sending early media, then send a second provisional response without SDP and with "inactive" in the *P-Early-Media* header.

If PEM support is disabled, or if it is enabled but the terminating endpoint does not send a *P-Early-Media* header, then Cisco BroadWorks behavior with regard to early media transitions is described in section *3.15 Early Media Transitions*. In particular, Cisco BroadWorks may apply the RFC 3398 policy. However, if PEM support is enabled and the terminating endpoint sends a *P-Early-Media* header, then the RFC 3398 policy is disabled for the call.

When Cisco BroadWorks receives a *P-Early-Media* header from the terminating endpoint and decides that an early media transition is needed, it may provide Media Server ringback, depending on whether it believes there is a network element that supports the Gating Function and can provide ringback. If the initial INVITE request has a *P-Early-Media* header, then Cisco BroadWorks assumes a Gating Function presence and avoids Media Server ringback.

This Gating Function ringback scenario is depicted in the following call flow diagram. Device A sends an initial INVITE request that has a *P-Early-Media* header with "supported". Cisco BroadWorks interprets this header to indicate that there is a network element that supports the Gating Function and can provide ringback. Device B sends a 183 (Session Progress) response with SDP and *P-Early-Media* containing "sendrecv", which Cisco BroadWorks interprets to mean that Device B will provide early media (such as remote ringback). Later, Device B sends a 180 (Ringing) response with no SDP and *P-Early-Media* with "inactive", which Cisco BroadWorks interprets to mean that an early media transition is needed. Because the Gating Function is present, Cisco BroadWorks relays the 180 (Ringing) response with "inactive" in the *P-Early-Media* header. The network element that provides the Gating Function detects the transition and provides ringback.

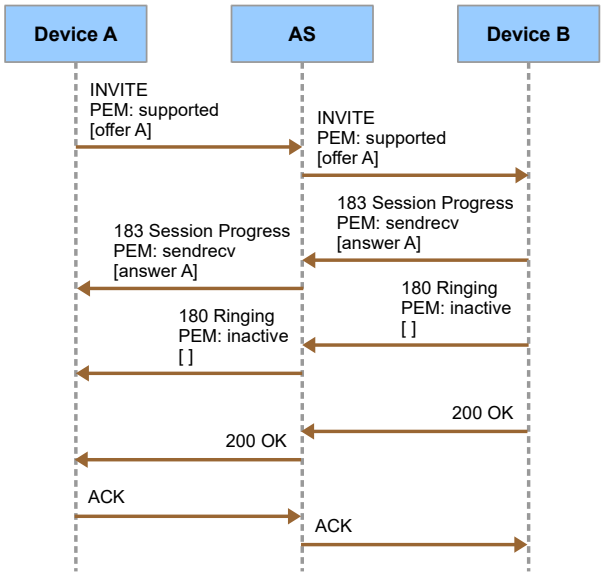

Figure 54 Early Media Transition with P-Early-Media and Gating Function Ringback

In an alternative scenario, the initial INVITE request does not have a *P-Early-Media* header and Cisco BroadWorks interprets this to mean that the Gating Function is absent. In this scenario, Cisco BroadWorks provides Media Server ringback for the transition.

The Media Server ringback scenario is shown in the following call flow diagram. Device A sends an initial INVITE request that omits a *P-Early-Media* header. Cisco BroadWorks interprets this omission to indicate that there is no Gating Function that is able to provide ringback. Device B sends a 183 (Session Progress) response with SDP and *P-Early-Media* with "sendrecv", which Cisco BroadWorks interprets to mean that Device B will provide early media (such as remote ringback). Later, Device B sends a 180 (Ringing) response with no SDP and *P-Early-Media* with "inactive", which Cisco BroadWorks interprets to mean that an early media transition is needed. Cisco BroadWorks relays the 180 (Ringing) response with "inactive" in the *P-Early-Media* header. Then, assuming a capable Gating Function is absent, Cisco BroadWorks provides Media Server ringback.

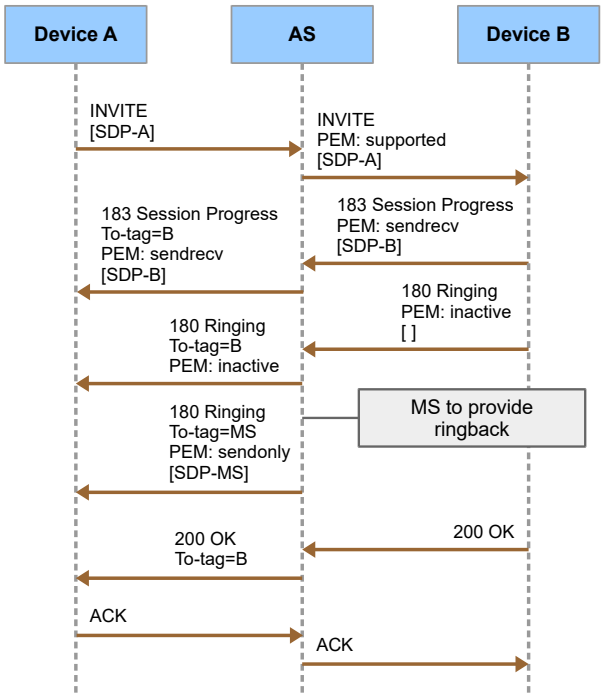

Figure 55 Early Media Transition with P-Early-Media and Media Server Ringback

3.35.3 Interactions With SIP Forking

3.35.3.1 Early Media Policy

When an initial SIP INVITE request forks to multiple terminating endpoints, the originating endpoint might receive early media from more than one source. A Policy Function correctly situated can implement policies that allow early media from a single source. Since Cisco BroadWorks tracks multiple early dialogs in the terminating session, Cisco BroadWorks is ideally situated to act as such a Policy Function.

The policies Cisco BroadWorks implements to select a single early media source can be complex. However, in a broad sense, the policies can be explained by two general principles. First, if Cisco BroadWorks is responsible for the forking, due to a user service such as Shared Call Appearance, then Cisco BroadWorks selects the early media source from the primary device endpoint. Cisco BroadWorks does not allow early media from any secondary device endpoint. Second, if a downstream proxy server is responsible for the forking, then Cisco BroadWorks normally assumes that the most recently created dialog is active and all older dialogs are inactive regarding early media. However, if Cisco BroadWorks receives a provisional response with a *P-Early-Media* that contains "sendrecv" or "sendonly", then it makes the associated dialog the active dialog, even if the dialog was previously changed to inactive. When a new dialog becomes active, Cisco BroadWorks sends a provisional response to block early media associated with the previously active dialog, if necessary. If the newly active dialog does not establish an early media stream (for example, if the provisional response does not have SDP, or if it has a *P-Early-Media* header with "inactive"), then Cisco BroadWorks may provide ringback tone via the Media Server, if necessary.

3.35.3.2 Cisco BroadWorks Forking Services

Cisco BroadWorks supports many forking services, such as Shared Call Appearance, BroadWorks Anywhere, and Simultaneous Ring. Cisco BroadWorks supports configuration options that enable Cisco BroadWorks either to consume or to relay provisional responses from the forked endpoints. Section *3.14 SIP Forking* provides introductory information about Cisco BroadWorks forking services and configuration.

When Cisco BroadWorks operates in a mode that consumes provisional responses from secondary endpoints, it modifies the SDP that it sends to those endpoints to become a "hold" SDP. In this way, Cisco BroadWorks attempts to prevent early media from secondary endpoints.

The following simplified call flow diagram shows how Cisco BroadWorks handles early media for a Shared Call Appearance scenario when it consumes provisional responses from the secondary endpoints. Device B is the Cisco BroadWorks user's primary device endpoint. Device B1 is the same user's secondary device endpoint. When Cisco BroadWorks forks the INVITE request to the secondary endpoint, it changes the SDP to a hold SDP (in this particular case, by changing the directionality to "sendonly"). When the secondary endpoint sends a provisional response, Cisco BroadWorks consumes the provisional response.

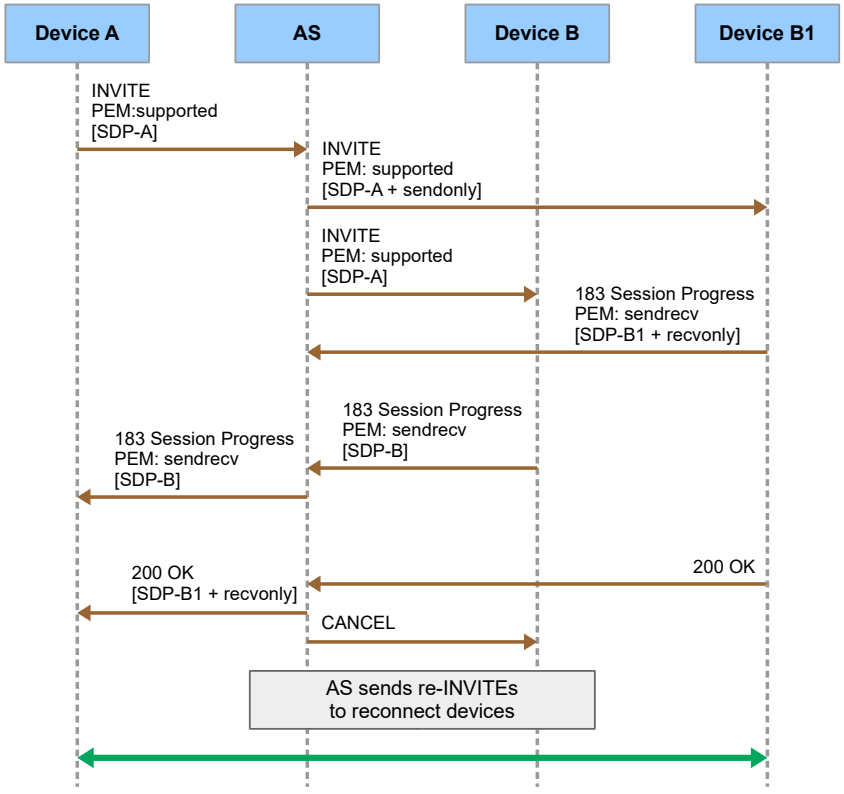

Figure 56 Shared Call Appearance with Application Server Consuming Provisional Responses from Secondary Endpoint

When Cisco BroadWorks operates in a mode that relays the provisional responses, Cisco BroadWorks allows early media only from the primary device endpoint. Thus, when Cisco BroadWorks relays a provisional response from a secondary device endpoint, it adds or modifies the *P-Early-Media* header to contain "inactive", which causes the Gating Function to block any early media from the secondary device endpoint.

The following simplified call flow diagram shows how Cisco BroadWorks handles early media for a Shared Call Appearance scenario. In this scenario:

- The SIP parameter *proxyForkingProvisionalResponses* is set to "true".
- The SIP parameter *supportPEarlyMediaHeader* is set to "true".
- Cisco BroadWorks operates in multiple dialog mode on the network interface toward the originating endpoint. (This implies the SIP parameter *networkForkingSupport* is set to "multipleDialogs". For more information, see section *3.14 SIP Forking*.)

Device A is a network device endpoint (that is, accessed via the network interface). Device B is a Cisco BroadWorks user's primary device endpoint. Device B1 is the same Cisco BroadWorks user's secondary device endpoint (a Shared Call Appearance device endpoint). Cisco BroadWorks relays the provisional response from the secondary device endpoint (Device B1); however, it sets the P-Early-Media header to "inactive". The PEM value causes the Gating Function to block early media from the secondary device endpoint. When Cisco BroadWorks relays the provisional response from the primary device endpoint (Device B), it permits the early media and relays the P-Early-Media header with "sendrecv".

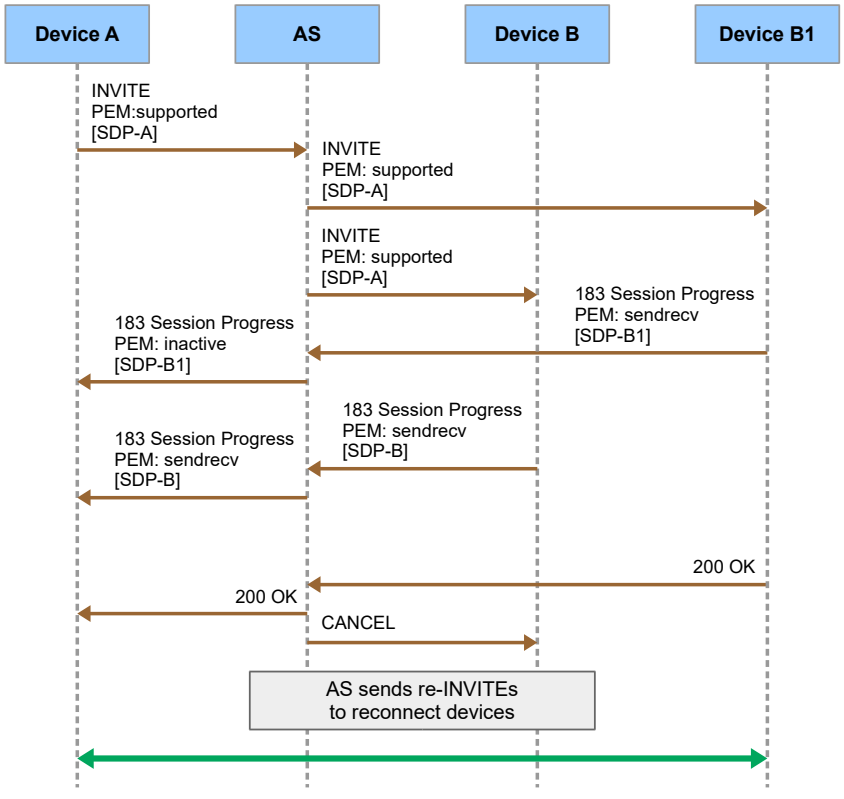

Figure 57 Shared Call Appearance with Application Server Relaying Provisional Responses from Secondary Endpoint

3.35.3.3 Proxy Server Forking

When Cisco BroadWorks sends an initial INVITE request, a downstream proxy server may fork that INVITE request, so that Cisco BroadWorks receives provisional responses for multiple early dialogs. In this scenario, Cisco BroadWorks early media policy selects one dialog to be the active dialog, setting all other dialogs to inactive. Cisco BroadWorks allows early media only from the active dialog. When Cisco BroadWorks transitions a dialog from active to inactive, if necessary it sends a new provisional response to the originating endpoint with *P-Early-Media* set to “inactive”.

To select the active dialog, Cisco BroadWorks operates as follows:

- If Cisco BroadWorks receives a provisional response that creates a new (early) dialog, it makes that dialog the active dialog.
- If Cisco BroadWorks receives a provisional response with a *P-Early-Media* header that contains “sendrecv” or “sendonly”, then it makes the associated dialog the active dialog, even if it’s an existing dialog.

The following simplified call flow diagram shows sequential forking by a downstream proxy server. Cisco BroadWorks operates in multiple dialog mode toward Device A. When the proxy server forks the INVITE request, Cisco BroadWorks correctly creates distinct dialogs. When Cisco BroadWorks receives the first provisional response from the proxy server, it creates a dialog and makes it the active dialog. When Cisco BroadWorks receives the second provisional response, it creates a new dialog, makes the old dialog inactive, and makes the new dialog active. When the old dialog transitions from active to inactive, Cisco BroadWorks sends a new provisional response with *P-Early-Media* set to “inactive”, which signals to the upstream Gating Function to block any early media from Device B1. Cisco BroadWorks then sends a provisional response with *P-Early-Media* set to “sendrecv” to allow early media from Device B2.

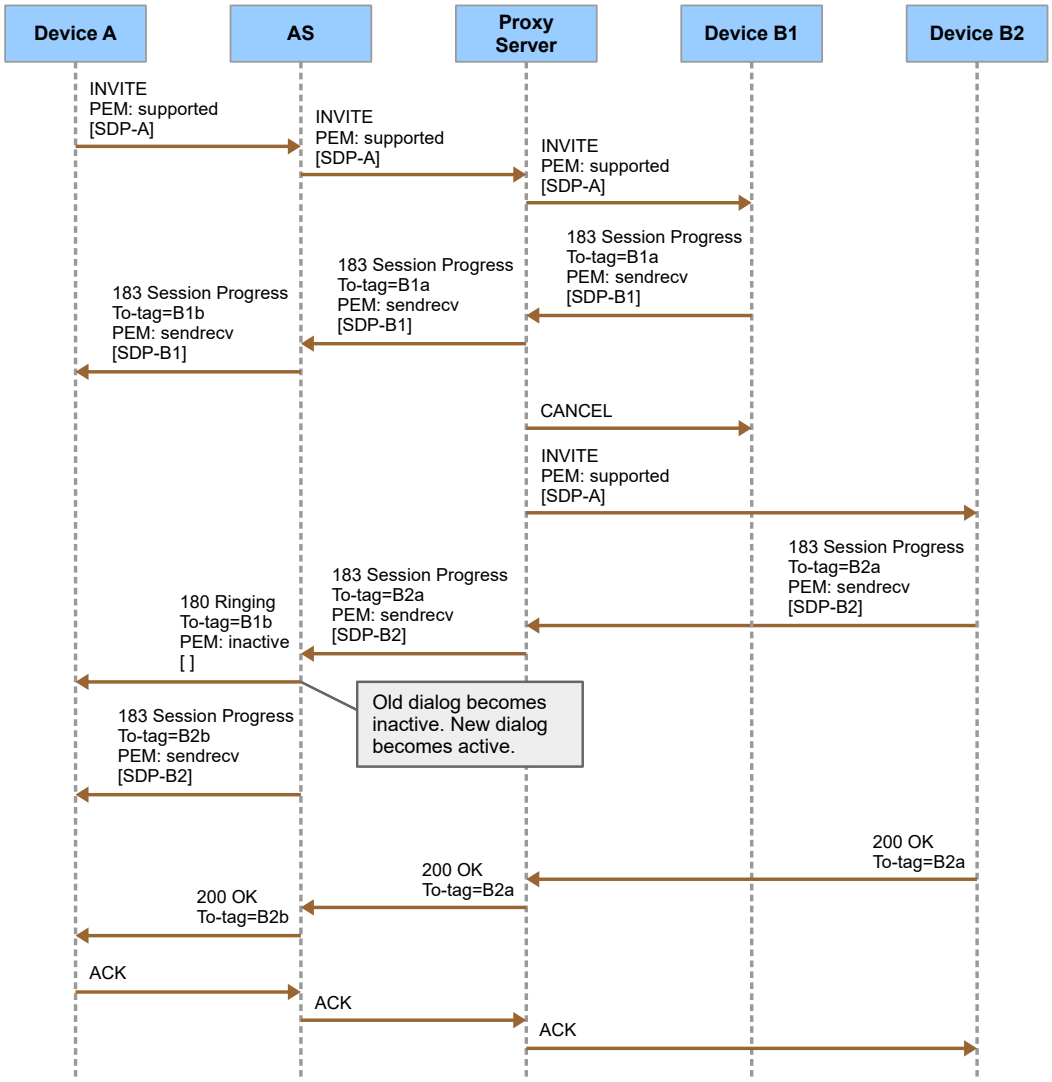

Figure 58 Proxy Server Forking and Early Media Source Selection

When Cisco BroadWorks creates a new dialog due to forking, it may need to provide ringback tone. In one such scenario, Cisco BroadWorks initially selects an active dialog as an early media source. When Cisco BroadWorks later creates a new dialog, it makes the newer dialog active and the older dialog inactive. If there is no early media associated with the new active dialog (because the provisional response contains no SDP, or because the provisional response has *P-Early-Media* with the value "inactive"), then Cisco BroadWorks provides ringback tone to the originating endpoint. Cisco BroadWorks may provide the ringback itself, via the Media Server, or it may depend on an upstream Gating Function to provide the ringback (or the actual originating endpoint may provide local ringback). This decision depends on the content of the initial INVITE request from the originating endpoint. If the INVITE request contained a *P-Early-Media* header, then Cisco BroadWorks depends on an upstream Gating Function to provide ringback tone. Otherwise, if the INVITE request omitted a *P-Early-Media* header, then Cisco BroadWorks provides Media Server ringback tone.

The following simplified call flow diagram shows a scenario in which Cisco BroadWorks assumes the presence of an upstream Gating Function that can provide ringback tone. The downstream proxy server forks the initial INVITE request to Device B1 and to Device B2. When Cisco BroadWorks receives a provisional response from Device B1, it creates a new active dialog and accepts Device B1 as the initial early media source. When Cisco BroadWorks receives a provisional response from Device B2, it sets the older dialog to inactive and creates a new active dialog. The provisional response from Device B2 indicates that the device will not provide early media. Therefore, Cisco BroadWorks decides that Device A should receive ringback tone. Because the INVITE request from Device A contained a *P-Early-Media* header, Cisco BroadWorks decides to let an upstream Gating Function provide ringback tone. (Alternatively, Device A itself could provide local ringback.)

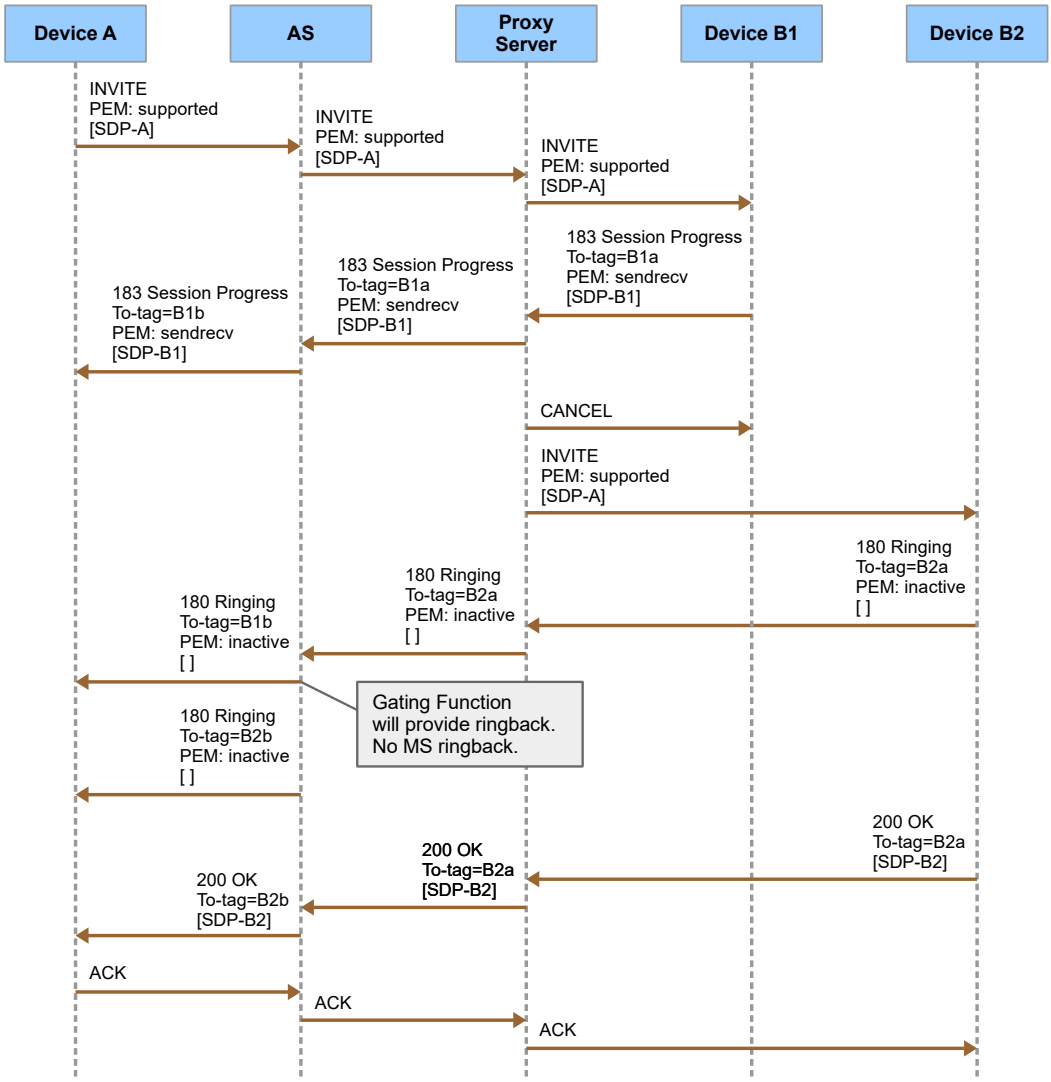

Figure 59 Proxy Server Forking with Gating Function Ringback

The following simplified call flow diagram shows the same scenario as the preceding one, except that the INVITE request from Device A has no *P-Early-Media* header. Consequently, Cisco BroadWorks sends an additional provisional response to Device A with Media Server SDP. This provisional response establishes a new early dialog for ringback tone provided by the Media Server.

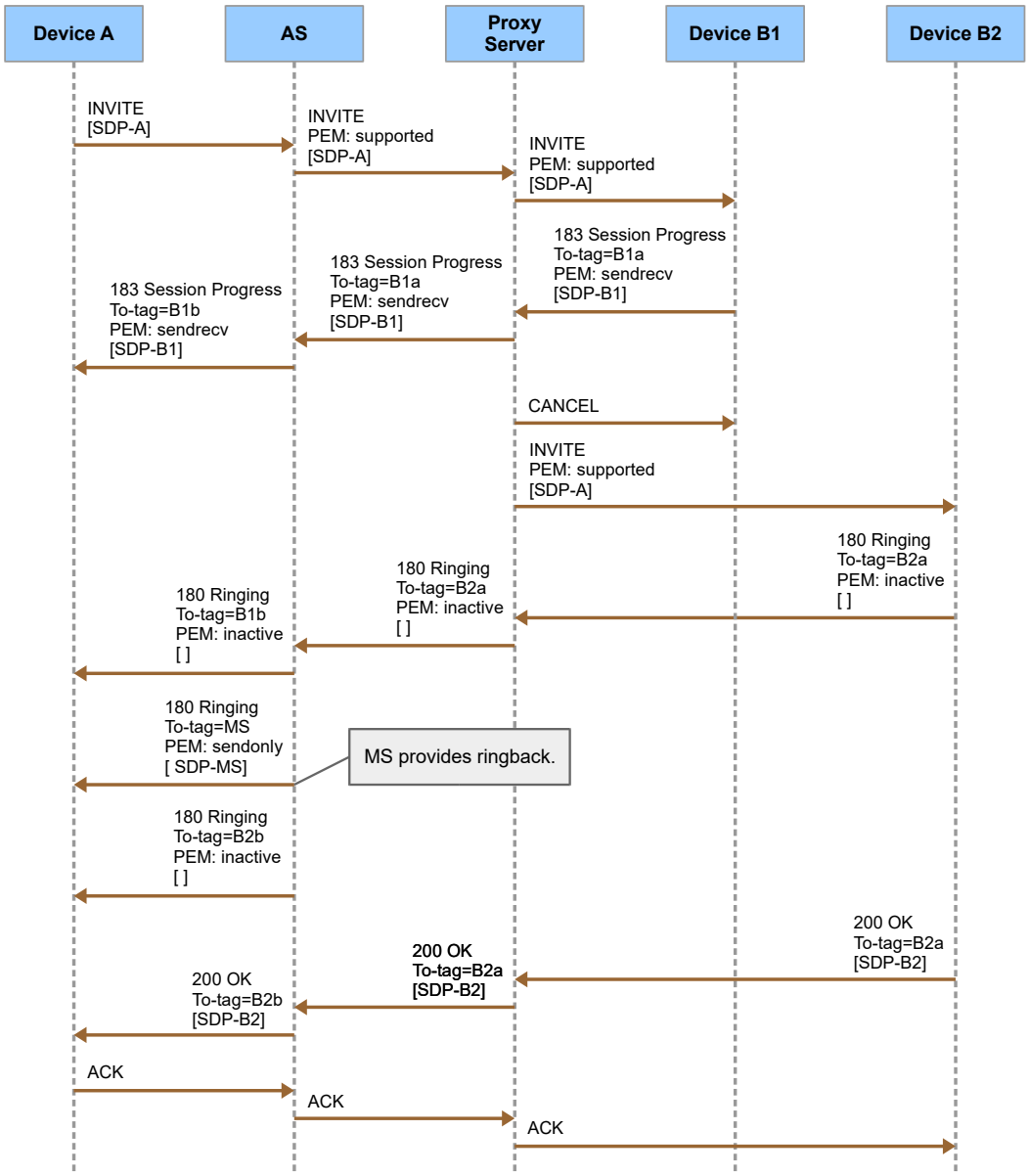

Figure 60 Proxy Server Forking with Media Server Ringback

3.36 Configurable Treatments and Reason Header (RFC 3326)

3.36.1 Treatments

A treatment is defined as the sequence of actions a telephony system takes when a call fails or is blocked. Typically, the telephony system plays a media announcement to the calling party, which indicates that the call could not complete as well as the reason why the call failed. Treatments also include the SIP signaling used to terminate the call, the *Reason* header, (see *RFC 3326* [44]) and the fields used in the call detail records.

Cisco BroadWorks defines a set of system-default treatments that apply to the different call failure or call blocking scenarios. In addition to these system-default treatments, a set of configurable treatments can be defined to override certain system behavior in specified scenarios.

The details of configurable treatments are provided in the *Cisco BroadWorks Treatment Guide* [45]. The following are some notable points relevant to the SIP interface:

- When Cisco BroadWorks receives a SIP error response to an initial INVITE request, the actions it takes are configurable. Cisco BroadWorks can apply a treatment based on:
  - the SIP response code
  - a *Reason* header with a Q.850 code or a SIP code
- When Cisco BroadWorks handles a failure condition or a blocked call, the response it sends to the initial INVITE request is configurable. Cisco BroadWorks can be configured to send:
  - a specified SIP response code
  - specified SIP response text
  - a *Reason* header with a specified Q.850 code and text
  - a *Reason* header with a specified SIP code and text
- When Cisco BroadWorks handles a failure condition or a blocked call, it can optionally play an announcement to the caller. If it is configured to play an announcement, it can:
  - play the announcement before answer using early media
  - or, play the announcement after answer using regular media

  If Cisco BroadWorks plays the announcement before answer, then it always sends a 487 response to the INVITE request after the announcement. The intention is that any SIP element receiving the 487 response should avoid playing a second announcement.
- Regardless of how treatments are configured, Cisco BroadWorks always handles a received 487 response specially. Cisco BroadWorks assumes the SIP element that sent the 487 response already played an announcement (or did not play an announcement and does not want any other network element to play an announcement). Therefore, when handling the *487* response, Cisco BroadWorks always avoids playing an announcement.
- Upon receiving certain SIP responses, Cisco BroadWorks may route advance. For example, Cisco BroadWorks may route advance after it receives a *503* response. The SIP response codes that trigger route advance are configurable.

- Cisco BroadWorks does not proxy a *Reason* header. However, it is possible to configure Cisco BroadWorks so that it appears to proxy the *Reason* header, with some limitations. When so configured, Cisco BroadWorks actually maps the incoming response to a treatment, and then maps the treatment to an outgoing response.

3.36.2 Reason Header

3.36.2.1 Syntax

The *Reason* header is defined in *RFC 3326*. Cisco BroadWorks supports the following syntax, which is compatible with the syntax in *RFC 3326*.

```
Reason = "Reason" HCOLON reason-value *(COMMA reason-value)
reason-value = protocol *(SEMI reason-params)
protocol = "SIP" / "Q.850" / "broadworks" / "BW-NS" / "Diversion"
           / token
reason-params = protocol-cause / reason-text / "no-recon-on-answer"
           / "reconnecting" / "xfer-cc-back" / "xfer-cc-front"
           / "bw-internal-forbidden" / "sac-group-rejection"
           / "sac-group-orig-rejection" / "sac-group-term-rejection"
           / "ea-call-push" / treatment / reason-extension
protocol-cause = "cause" EQUAL cause
cause = 1*DIGIT
reason-text = "text" EQUAL quoted-string
treatment = "treatment" EQUAL quoted-string
reason-extension = generic-param
```

Compared to the *RFC 3326* syntax, the Cisco BroadWorks syntax adds additional alternatives to the protocol definition and the *reason-params* definition. The added protocol alternatives, "broadworks", "BW-NS", and "Diversion", are described in the following subsections.

3.36.2.2 SIP Protocol

Cisco BroadWorks supports the "SIP" protocol in the *Reason* header via configurable treatments.

3.36.2.3 Q.850 Protocol

Cisco BroadWorks supports the "Q.850" protocol in the *Reason* header via configurable treatments.

3.36.2.4 Cisco BroadWorks Protocol

Cisco BroadWorks uses the "broadworks" protocol in the *Reason* header to improve the operation of certain call processing operations. When using the "broadworks" protocol, Cisco BroadWorks adds a parameter to indicate a specific condition, as described below.

- *no-recon-on-answer*, *reconnecting* – Cisco BroadWorks uses these parameter values to avoid a glare condition when it needs to immediately send a re-INVITE request after receiving a 200 response to the current INVITE request. The Cisco BroadWorks server that sends one of these parameters will attempt a reconnect operation. The Cisco BroadWorks server that receives one of these parameters will avoid a reconnect operation.

  Examples:

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.16.145.11:5060;branch=z9hG4bK-87e9f017
```

```
From:"John Q.
Public"<sip:+12145550000@broadworks.test>;tag=792fed98de92e76fo0
To:<sip:5125550102@broadworks.test>;tag=470121785-1374682408095
Call-ID:1cd6d38c-1d9e1a0b@10.16.145.11
Cseq:101 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
Supported:
Contact:<sip:10.16.145.3:5060>
P-Asserted-Identity:"Dom
Portwood"<sip:+15125550102@initech.test;user=phone>
Privacy:none
Reason:broadworks;no-recon-on-answer
Accept:application/media_control+xml,application/sdp,application/x-
broadworks-call-center+xml
Content-Type:application/sdp
Content-Length:167
...
```

```
ACK sip:102a@10.16.145.6:5060 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.6V5060-0-146434167A1134805734-1374682403053-
From:"John Q.
Public"<sip:2145550000@initech.test;user=phone>;tag=1134805734-
1374682403053-
To:"Dom Portwood"<sip:102a@initech.test>;tag=1115500032
Call-ID:BW111323053240713-1240638653@10.16.145.3
Cseq:146434167 ACK
Contact:<sip:10.16.145.3:5060>
Reason:broadworks;reconnecting
Max-Forwards:10
Content-Length:0
```

- *xfer-cc-back* and *xfer-cc-front* – Cisco BroadWorks sends one of these parameters when a Call Center agent transfers a call back to the Call Center to be re-queued.

  Example:

```
Reason:broadworks;xfer-cc-front
```

- "bw-internal-forbidden" – Cisco BroadWorks may include this parameter when it sends a failure response back to a Hunt Group on behalf of a Hunt Group agent. This parameter signals that the agent should be considered reachable when it would otherwise be considered unreachable.

  Example:

```
SIP/2.0 403 Forbidden
Via:SIP/2.0/UDP 10.16.135.49;branch=z9hG4bKBroadWorks.1lp1imq-
10.16.135.49V5060-0-169308048-1281438224-1308156157727-
From:"john2
north"<sip:9726987502@10.16.135.49;user=phone;otg=otg>;tag=1281438224-
1308156157727-
To:<sip:+19726987500@10.16.135.49:5060;user=phone>;tag=697508725-
1308156158033
Call-ID:BW114237727150611253088241@10.16.135.49
Cseq:169308048 INVITE
Reason:broadworks;bw-internal-forbidden
Content-Length:0
```

- *sac-group-rejection*, *sac-group-orig-rejection*, and *sac-group-term-rejection* – Cisco BroadWorks sends one of these reasons to indicate that it blocked a call because of Session Admission Control.

  Example:

```
Reason:broadworks;sac-group-orig-rejection
```

- *ea-call-push* – Cisco BroadWorks adds this reason to a BYE request when executing and Executive-Assistant Call Push operation.

  Example:

```
Reason:broadworks;ea-call-push
```

3.36.2.5 BW-NS Protocol

Cisco BroadWorks uses this *Reason* protocol only in SIP messages exchanged between the Application Server and the Network Server. The protocol allows the Network Server to convey translations-related information to the Application Server for use in configurable treatments.

Example:

```
SIP/2.0 403 Forbidden
Via:SIP/2.0/UDP 192.168.8.101:5061;branch=z9hG4bK-16c6c870
From:"userA"<sip:5146993604@mtlasdev93.net>;tag=4cf29974133eedd0
To:<sip:603@mtlasdev93.net>;tag=121156610-1164902613031
Call-ID:67b7ed44-4ced2c40@192.168.8.101
Cseq:101 INVITE
Reason:BW-NS;treatment="invld"
Content-Length:0
```

3.36.2.6 Diversion Protocol

When Cisco BroadWorks redirects a call, it can optionally return a *181* response. The *181* response contains a *Reason* header with the "Diversion" protocol.

The following is an example of the SIP *181* response.

```
SIP/2.0 181 Call is being forwarded
From:"redirected User"<sip:redirected@example.net:5060>;tag=46b61cd86a4
To:<sip:redirecting@example.net:5060>;tag=808254021-1254254996796
Call-ID:73915207-de22f3cb@example.net
Cseq:101 INVITE
Supported:
Contact:<sip:example.net:5060>
Rseq:62273182
P-Asserted-Identity:<sip:redirecting@example.net>
Privacy:none
Reason:Diversion;text="busy"
```

3.37 Connected Line Identification Presentation

The Connected Line Identification Presentation (COLP) service provides the calling party with the ability to be presented with the identity of the connected party, which may or may not be the dialed party.

3.37.1 Cisco BroadWorks Sending COLP

When COLP is provided by Cisco BroadWorks, it is sent in *18x* and *200 OK* responses to the initial INVITE requests, as well as UPDATE and re-INVITE requests.

The header used to provide COLP depends on the setting of the *privacyVersion SIP* system parameter. Note that IP Multimedia Subsystem (IMS) mode and distributed group call (DGC) signaling always function as if the *privacyVersion* system parameter is set to "RFC 3323".

- If the *privacyVersion* system parameter is set to "RFC 3323", then COLP is provided via the *P-Asserted-Identity* header.
- If the *privacyVersion* system parameter is set to "privacy-03" or "privacy-00", then COLP is provided via the *Remote-Party-ID* header.
- If the *privacyVersion* system parameter is set to any other value, then COLP is not provided in the SIP messages.

The COLP included in the *PAI/RPID* header is always a SIP URI entry. If the COLP is an E.164 phone number, is being sent to a network device when the sendE164 SIP system parameter is set to "false", and the phone number's country code matches the system country code, then the phone number is normalized to the appropriate prefixed national format for the country code. Otherwise, the COLP is sent without normalization.

The current COLP for a call is always provided in the message sent to a network subscriber based on the *privacyVersion* system parameter setting.

3.37.2 Cisco BroadWorks Receiving COLP

Cisco BroadWorks accepts COLP updates in *18x* and *200 OK* responses to the initial INVITE requests, as well as UPDATE and re-INVITE requests.

COLP can be received in the *P-Preferred-Identity (PPI), PAI, or RPID* header of an *18x* or *200 OK* response. The headers are treated in precedence order as listed, so the *PPI* header always has the highest precedence. Note that the presence of the *PPI* header should be rare (the *PAI* header is usually expected to be present), but the *PPI* header is allowed and has precedence to maintain consistency with how the headers are processed for CLID purposes in an INVITE. Also note that the setting of the *privacyVersion* system parameter has no effect on receiving COLP.

COLP received for a network subscriber is always accepted and used. Note that the COLP can be normalized to E.164 and only the *phone-context* and *user=phone* parameters are processed.

If a *PPI* or *PAI* header is received, then if present, the tel uniform resource identifier (URI) entry is always used as the COLP. If there is not a tel URI entry, then the SIP URI entry (if present) is used as the COLP instead.

3.38 External Custom Ringback

The External Custom Ringback service allows Cisco BroadWorks to contact an external server to provide Custom Ringback functions to subscribers. It builds on Cisco BroadWorks Custom Ringback function by substituting the INVITE to the Cisco BroadWorks Media Server, to an INVITE to a specific directory number (DN), to reach the External Custom Ringback server, which provides the media to be returned to the caller.

The business logic to select the ringback media is done entirely by the External Custom Ringback server, based on the calling and called parties, as well as other factors such as the date and time.

The External Custom Ringback service operates as follows:

1) When a Cisco BroadWorks user receives an incoming call, the External Custom Ringback service determines whether it should process this call.
2) If yes, the Application Server contacts the External Custom Ringback server.
3) The external custom ringback server responds and provides the ringback media to the caller.
4) The connection to the external custom ringback server is released when the call is answered or released.

3.38.1 SIP INVITE to External Custom Ringback

Once the Application Server has determined that a connection to the external custom ringback server should be attempted, it initiates contact by sending a SIP INVITE to the External Custom Ringback server.

The calling party identity is provided in the SIP *From* header of the INVITE message sent to the custom ringback server. The server is considered trusted; the appropriate identity and privacy headers are included, as shown in the following table. The calling party identity is provided regardless of the calling line ID configuration. This aligns with Cisco BroadWorks custom ringback logic, which ignores calling line ID delivery settings.

AS_CLI/Interface/SIP privacyVersion	Included Headers
RFC3323 and IMS mode	*P-Asserted-Identity* *Privacy*
RFC3323-Japan	*P-Asserted-Identity* *Privacy (if CLID blocked)*
privacy-03	*Remote-Party-ID* *RPID-Privacy* *Proxy-Require:privacy*
privacy-00	*Remote-Party-ID* *Anonymity* *Proxy-Require:privacy*

The *Request-URI* sent to the external custom ringback server depends on the user configuration. If the user server SIP URI is selected, it is used in the outgoing INVITE. If the service provider settings are selected, the *Request-URI* is built as follows.

The user part of the *Request-URI* is composed of a prefix configured at the service provider level, appended with the user main DN (or group DN if no user DN is configured) in national format (that is, without the country code but with the national prefix). Note that a user who has neither a DN nor a group DN cannot receive calls, and thus External Custom Ringback does not apply. The host part of the *Request-URI* is constructed using the custom ringback server address and port configured at the service provider level. The user=phone parameter is also added. For example, the following configuration results in "INVITE sip:9992223334444@ringback.domain.net:5050;user=phone SIP/2.0".

User DN:	+12223334444
Prefix:	999
Custom Ringback Server Address:	ringback.domain.net
Custom Ringback Server Port:	5050

The *To* header contains the same location as the *Request-URI*.

3.38.2 External Custom Ringback Server Response

The external custom ringback server is responsible for providing the early media to the calling party. It can, however, provide it as early media, or as regular media, in which case the Application Server makes the appropriate signaling transformations to provide it as early media to the calling party.

- When the external custom ringback server provides ringback as early media, it responds to the initial INVITE message with an *18x* response containing an SDP with the appropriate media description. If the server uses reliable provisional responses (*RFC 3262*), the Application Server sends the appropriate PRACK message. However, the PRACK message is not passed end-to-end between the calling party and the external custom ringback server, and as a result, it cannot be used to carry an SDP.
- When the external custom ringback server provides ringback as regular media, it responds to the initial INVITE message with a *200* response containing an SDP with the appropriate media description. This *200* response can be preceded by one or more provisional responses without SDP. The Application Server uses the SDP from the *200* response to construct a provisional response for the calling party.

3.38.3 Media Changes

This subsection describes Cisco BroadWorks behavior when the external custom ringback media is renegotiated while it is playing. There are several cases to consider:

- If the external custom ringback server provides ringback media using a *200*, then most media renegotiation succeeds. The Application Server converts messages with SDP from the calling party, either PRACK or UPDATE, into a corresponding re-INVITE to the external custom ringback server. The reverse is true as well; the Application Server converts the re-INVITE from the external custom ringback server into UPDATE messages sent to the calling party, if allowed by the calling party.

NOTE: Re-INVITE without SDP and UPDATE with SDP from the external custom ringback server are not supported.

- If the external custom ringback server provides ringback media using provisional responses, then media renegotiation is not supported. PRACK or UPDATE with SDP from the calling party or the external custom ringback server does not succeed. In addition, it is not supported for the external custom ringback server to provide a different SDP in a subsequent *18x* or *200* responses.

Unsupported SDP negotiation attempts result in the calling party being connected to local Media Server ringback and the external custom ringback server being released.

3.38.4 Video Support

The media selection is negotiated between the calling party and the external custom ringback server. The exchanged SDP may or not contain video, and this has no impact on the External Custom Ringback service. Video is provided if both the calling party and the external custom ringback server support it.

3.39 AccessCode SIP Header

Cisco BroadWorks offers optional support of the SIP AccessCode header.

The Session Initiation Protocol (SIP) *AccessCode* header addresses the issue of service interaction between Internet Protocol (IP) Centrex services and non-IP Centrex services in next generation network (NGN) deployments. In these deployments, the softswitch invokes all the services in a preconfigured order for a particular call based on the response from the Smart Home Location Register (SHLR). When it is time to execute IP Centrex services, the softswitch sends an INVITE to the Application Server with an *AccessCode* header, and the Application Server proxies, replaces, or adds the *AccessCode* header based on the call scenario. The softswitch then executes the next service based on the *AccessCode* header returned by the Application Server.

3.39.1 Header Syntax

The *AccessCode* header can be included in initial SIP INVITE messages.

The *AccessCode* header has the following syntax.

```
AccessCode = "AccessCode" HCOLON gen-value
```

The following is an example of the *AccessCode* header.

```
AccessCode:1234
```

3.39.2 Originations

When this feature is enabled, the *AccessCode* header received in an initial INVITE from an access device for a user origination is proxied into the outgoing INVITE for the origination if it is sent to the network. If no *AccessCode* header is present in the INVITE received for the origination, then the INVITE sent to the network for the origination has the *AccessCode* header added using the value configured for the *redirectingAccessCode* system parameter.

Also note that the *AccessCode* header is ignored if it is received from the network, or if this feature is disabled.

3.39.3 Redirections

When this feature is enabled, the *AccessCode* header is included in all initial INVITEs sent to the network for a user redirection (for example, Call Forwarding, Simultaneous Ringing, and so on). The *AccessCode* header is set to the value configured for the *redirectingAccessCode* system parameter.

BroadWorks Anywhere and Remote Office terminations are sent to the network and treated as user redirections. Therefore, the value configured for the *redirectingAccessCode* system parameter should also be used.

When this feature is disabled, the *AccessCode* header is not included in an INVITE sent for a user redirection.

3.39.4 Click-to-Dial Calls

When this feature is enabled, the *AccessCode* header is included in the initial INVITE sent for the first leg of the Click-to-Dial call. The *AccessCode* header is set to the value configured for the *clickToDialAccessCode* system parameter.

Note that this functionality applies to both regular Click-to-Dial calls that send the INVITE to an access device, and to BroadWorks Anywhere and Remote Office Click-to-Dial calls that send the INVITE to the network.

Once the first leg of the Click-to-Dial call has been answered, the initial INVITE sent for the second leg of the Click-to-Dial call also includes the *AccessCode* header. If the INVITE is sent to the network, then the *AccessCode* header is set to the value configured for the *redirectingAccessCode* system parameter.

When this feature is deactivated, the *AccessCode* header is not included in an INVITE sent for a Click-to-Dial call.

3.40 Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP) (RFC 3455, RFC 7315)

RFC 7315 describes several SIP header fields that were introduced for use in 3GPP standards. In IMS mode, Cisco BroadWorks' support for these headers is described in the *Cisco BroadWorks AS Mode ISC Interface Specification*. This section describes Cisco BroadWorks support for these headers in standalone mode (that is, non-IMS mode).

For backward compatibility, Cisco BroadWorks supports the syntax of *RFC 3455* as well as the syntax of *RFC 7315*.

3.40.1 P-Called-Party-ID Header

The Cisco BroadWorks Application Server supports the *P-Called-Party-ID* SIP header defined in *RFC 7315*.

- For a Cisco BroadWorks user origination, the Application Server proxies the *P-Called-Party-ID* header in an initial INVITE in non-IMS deployments.
- For a Cisco BroadWorks user termination, the Application Server proxies the *P-Called-Party-ID* header in an initial INVITE if the destination is the user's primary location. It does not proxy the *P-Called-Party-ID* header to the user's secondary or alternate locations.

Example header:

```
P-Called-Party-ID: sip:user1-business@example.com
```

3.40.2 P-Access-Network-Info Header

Cisco BroadWorks accepts the *P-Access-Network-Info* header from an INVITE request from a network device and passes its value to the terminating session.

Cisco BroadWorks adds the *P-Access-Network-Info* header to an INVITE request to a network device if it received a value from the originating session.

3.40.3 Other Headers

3.40.3.1 P-Charging-Function-Addresses Header

Cisco BroadWorks recognizes the *P-Charging-Function-Addresses* header but does not proxy it in standalone mode.

3.40.3.2 P-Charging-Vector Header

Cisco BroadWorks recognizes the *P-Charging-Vector* header but does not proxy it in standalone mode.

3.40.3.3 P-Associated-URI Header

Cisco BroadWorks does not recognize the *P-Associated-URI* header.

3.40.3.4 P-Visited-Network-ID

Cisco BroadWorks does not recognize the *P-Visited-Network-ID* header.

3.41 Via Header

Cisco BroadWorks constructs the *Via* header according to rules specified in *RFC 3261*. Two cases must be considered. In the usual case, the branch parameter is constructed by appending the following components separated by "-" characters:

- The prefix "z9hG4bKBroadWorks."
- An encoded hash value representing the host address
- The destination IP and port separated by a V
- An internal index associated with the destination
- The message sequence number and the *From* header tag separated by a "-" or an "A" for ACK of INVITE 2xx responses.

An example of this is as follows.

```
Via:SIP/2.0/UDP 192.168.8.249;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.8.28V5060-0-1027284258-881047944-1232562698818-
```

… where "z9hG4bKBroadWorks." Is the prefix, "1su2iau" is a hash of the server address, "192.168.8.28V5060" is the destination IP and port, "0" is an internal index, "1027284258" is the Cseq number, and "881047944-1232562698818-" is the *From* header tag.

3.42 Automatic Callback

Cisco BroadWorks supports two signaling standards for automatic callback (ACB) functionality. The first, referred to as Call Completion Services here, is based on [48] *draft-poetzl-sipping-call-completion-02.* The second, referred to as Legacy Automatic Callback, is based on [53].

3.42.1 Call Completion Services

Cisco BroadWorks implements a subscription service for compliance with the MMTEL Completion of Communications to Busy Subscribers (CCBS) service. This subscription service is based on Internet Draft "*Extensions to the Session Initiation Protocol (SIP) for the support of the Call Completion Services for the European Telecommunications Standards Institute" (draft-poetzl-sipping-call-completion-02)* [48].

This Call Completion service is used to provide Automatic Callback between an Application Server and another network element (typically another Application Server). The originating Application Server first subscribes to the call-completion package. The terminating Application Server then queues the subscription and starts monitoring the requested user. Once the monitored user is deemed available for callback, the terminating Application Server notifies the originating Application Server, which triggers the recall.

NOTE: The draft used is expired. To prevent any potential interoperability issues with network elements implementing a newer version of the draft, the name of the event package is changed to *bw-call-completion*.

Following are some SIP message flow diagrams that show this mechanism. *Figure 61* shows how an inter-Application Server call can be identified as an Automatic Callback potential call, and how the originating Application Server subscribes to the *bw-call-completion* event package. *Figure 62* shows how the callback is triggered once the terminating user becomes available. *Figure 63* shows a successful callback and subscription termination.

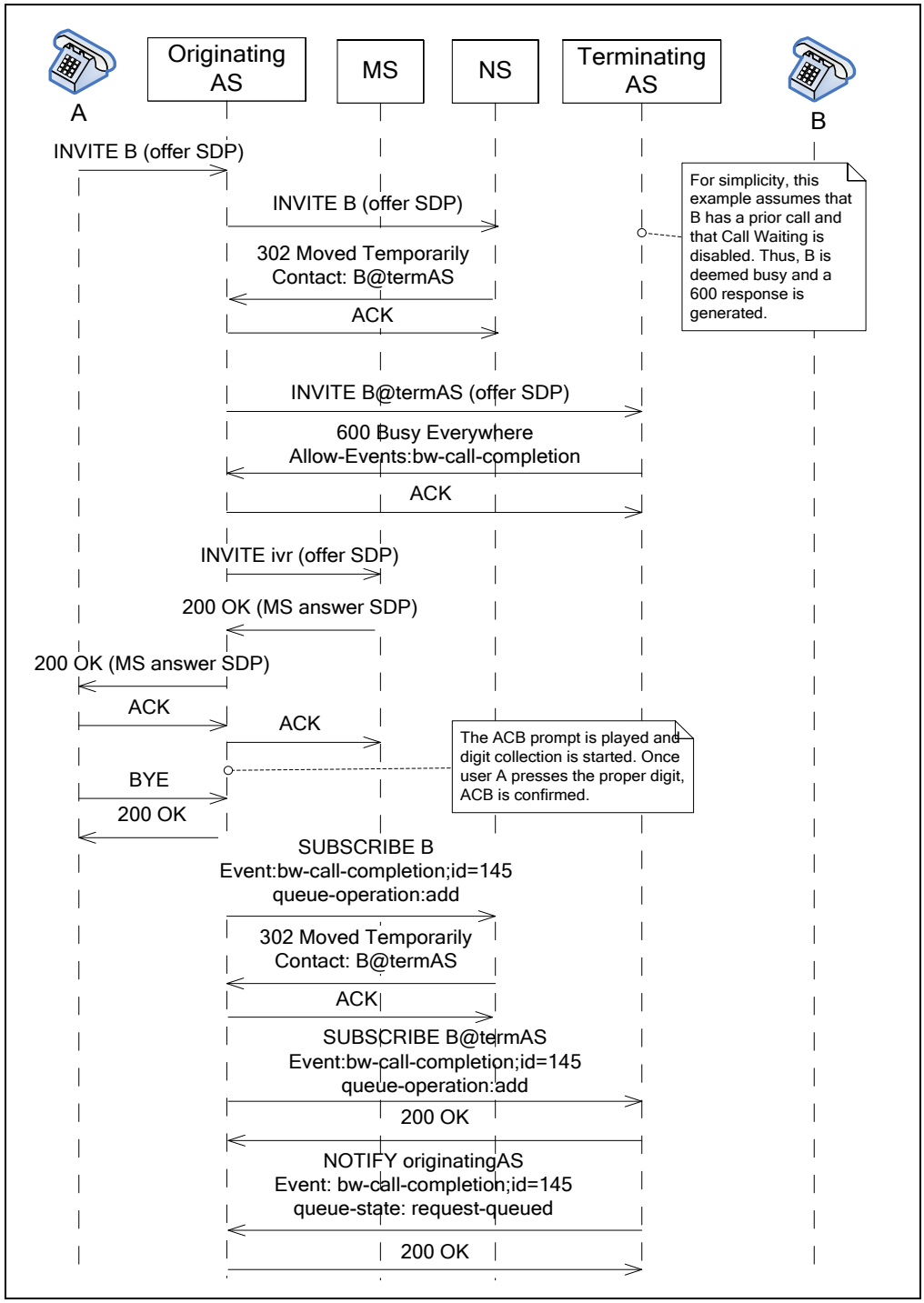

Figure 61 Inter-Application Server Initial Busy Call and Automatic Callback Monitoring Setup

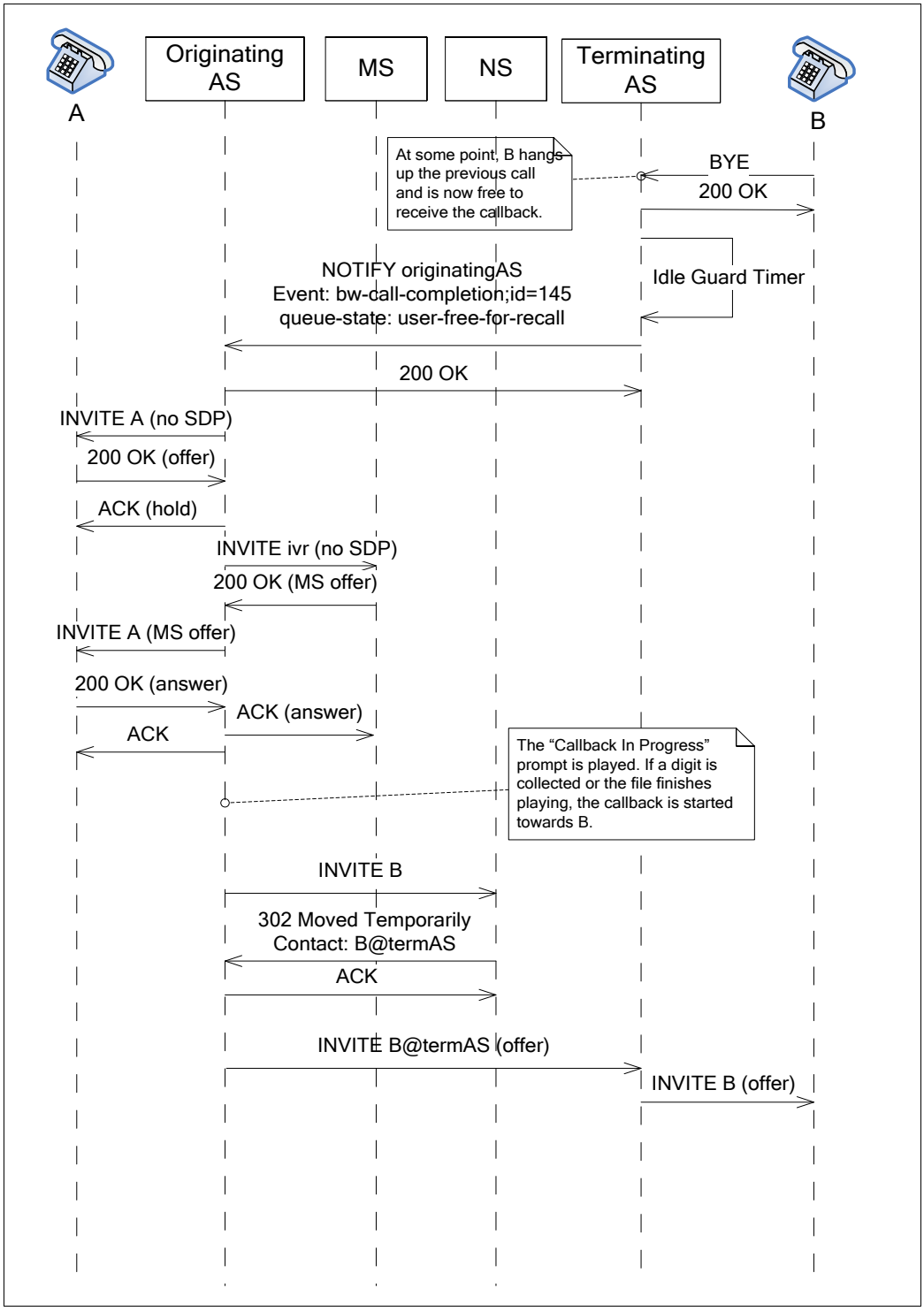

Figure 62 Inter-Application Server Callback Triggered and Recall Initiation

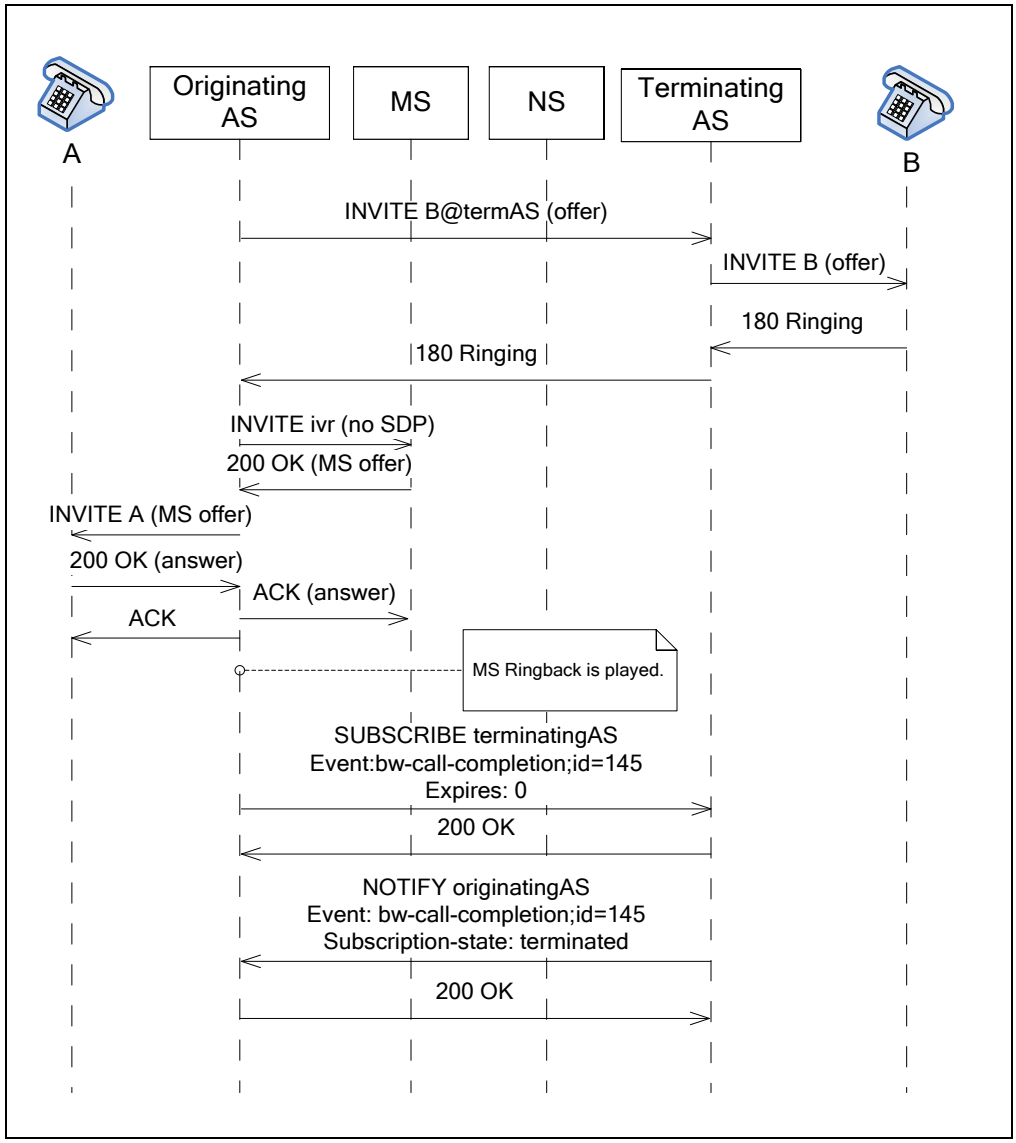

Figure 63 Inter-Application Server Callback Successful and Subscription Terminated

3.42.1.1 Allow-Events Header in SIP Error Responses

To indicate support for the *bw-call-completion* event package, the Cisco BroadWorks Application Server includes the *Allow-Events* header with the value “bw-call-completion” in negative SIP responses to INVITE requests when automatic callback is allowed. These are typically 600 Busy Everywhere responses when signaling busy conditions toward the network and 606 Not Acceptable responses when terminating users are not registered.

If Cisco BroadWorks receives a negative SIP response without the *Allow-Events* header or without the “bw-call-completion” value, then no attempt to SUBSCRIBE to the event package is made.

The following is an example of a negative SIP response with a valid *Allow-Events* header for Call Completion.

```
SIP/2.0 600 Busy everywhere
Via:SIP/2.0/UDP 192.168.8.24;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.8.24V5060-0-1031781628-136252873-1202506922487-
```

```
From:"userA"<sip:5146993604@192.168.8.24;user=phone>;tag=136252873
To:<sip:+15146959662@192.168.8.24:5060;user=phone>;tag=427176647
Call-ID:BW164202487080208282623023@192.168.8.24
Cseq:1031781628 INVITE
Allow-Events: bw-call-completion
Content-Length:0
```

3.42.1.2 bw-call-completion SUBSCRIBE Messages

The bw-call-completion event package, similar to other event packages, follows *RFC 3265* for the basic SIP SUBSCRIBE/NOTIFY exchange. The event package name is "bw-call-completion" and it is used in the *Event* and *Allow-Event*s header, as well as in the *event* parameter in the *Via* header.

Additionally, the bw-call-completion event package requires a specific body type to be present in SUBSCRIBE messages. This event body is formatted in the "application/bw-call-completion" data format. The bw-call-completion format is specified as a series of attribute value pairs defined as "attribute: value"; each pair is specified on a separate line.

- For SIP SUBSCRIBE requests, Cisco BroadWorks supports four attribute values. The first is *queue-nature*. This attribute gives the possibility to specify multiple queue types. Cisco BroadWorks only supports "CCBS", so the *queue-nature* value is always equal to "CCBS" in SUBSCRIBE requests generated by the originating side.
- The second supported attribute is *queue-operation*. This attribute has three possible values: "add", "suspend", and "resume". The "add" value is specified on initial SUBSCRIBE requests to indicate that the new subscription should be added to the queue. The "suspend" value is used by the originator to tell the terminator that the originator is not currently available for callback, but would like to stay in the callback queue. The "resume" value is used by the originator to indicate that it is available for callback once again. The "suspend" and "resume" values are used by Cisco BroadWorks when the originator is deemed busy upon recall.
- The third supported SUBSCRIBE body attribute is *caller*. This attribute is set to the public SIP URI of the originator. It is used to uniquely identify the originator of the callback in the queue.
- The fourth and last supported attribute is *service-retention*. The presence of this attribute (without a value) indicates to the terminator that the originator would like to remain in the queue even if the terminator is deemed busy again upon callback. Cisco BroadWorks supports this option and does not automatically remove the subscription from the queue upon callback. The removal of an entry in the queue is the responsibility of the originator. When the originator deems that the callback has completed successfully (when an 18x or *200 OK* response is received for the callback INVITE), it sends an un-subscribe request to the terminator. This is done by sending a SIP SUBSCRIBE request with the *Expires* header set to "0".

Note that in initial SIP SUBSCRIBE requests, the *Expires* header is set to the configured monitoring time of the originator. The terminating side does not explicitly send a NOTIFY terminated message when the timer expires; however, the callback queue is cleaned up silently when the terminator becomes available once again.

As shown in *Figure 61*, the initial SUBSCRIBE request is first sent to the Network Server. The contacts returned by the Network Server in a 302 response are then used to populate and route the SUBSCRIBE request to the terminating side. The *Request-URI* and *To* headers sent to the Network Server contain the called number (similar to the INVITE request sent in the same scenario). Note that for a phantom user (user without DN, reachable through extension dialing only), the *Request-URI* and *To* headers contain the user's primary user-id (that is, username@domain). Upon routing to the terminating side, the *Request-URI* and *To* headers contain the contact information received from the Network Server. The *From* header for both the request sent to the Network Server and the request sent to the terminating Application Server contain the user's CLID. Privacy policies do not apply here because there is no way that the message ever reaches the terminating user. The *From* header is not used to identify the caller; the "caller" entry in the body is used instead. Note that re-SUBSCRIBE requests use the Contact received in the initial SUBSCRIBE response and as a result, they are not required to go to the Network Server again.

Following is an example of a SIP SUBSCRIBE request sent by the Application Server to the Network Server.

```
SUBSCRIBE sip: 59662@ns1.example.org;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.8.24;branch=z9hG4bKBroadWorks.-1su2iau-
;event=bw-call-completion
From: <sip:+15146993604@as1.example1.org;user=phone>;tag=1051252247-
To: <sip: 59662@ns1.example.org;user=phone>
Call-ID: BW164202487080208282623023@192.168.8.24
Cseq: 114591234 SUBSCRIBE
Contact:<sip:192.168.8.24:5060>
Max-Forwards: 10
Event: bw-call-completion;id=145
Expires: 108000
Accept: application/bw-call-completion
Content-Type: application/bw-call-completion
Content-Length: 95

queue-nature: ccbs
queue-operation: add
caller: <sip:5146993604@example1.org>
service-retention
```

Following is an example of an initial SIP SUBSCRIBE request sent by the originating Application Server to the terminating network element.

```
SUBSCRIBE sip:+15146959662@as2.example2.org;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.8.24;branch=z9hG4bKBroadWorks.-1su2iau-
;event=bw-call-completion
From: <sip:+15146993604@as1.example1.org;user=phone>;tag=1234567890-
To: <sip: +15146959662@as2.example2.org;user=phone>
Call-ID: BW123456789012345678901234@192.168.8.24
Cseq: 114594567 SUBSCRIBE
Contact:<sip:as1.example1.org:5060>
Max-Forwards: 10
Event: bw-call-completion;id=145
Expires: 108000
Accept: application/bw-call-completion
Content-Type: application/bw-call-completion
Content-Length: 95

queue-nature: ccbs
queue-operation: add
```

```
caller: <sip:5146993604@example1.org>
service-retention
```

Following is an example of a SIP re-SUBSCRIBE request used to terminate the subscription.

```
SUBSCRIBE sip:as2.example2.org:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.8.24;branch=z9hG4bKBroadWorks.-1su2iau-
;event=bw-call-completion
From: <sip:+15146993604@example1.org;user=phone>;tag=1234567890-
To: <sip:+15146959662@example2.org;user=phone>;tag=0987654321-
Call-ID: BW123456789012345678901234@192.168.8.24
Cseq: 114597890 SUBSCRIBE
Contact:<sip:as1.example1.org:5060>
Max-Forwards: 10
Event: bw-call-completion;id=145
Expires: 0
Content-Length: 0
```

Because the terminating server has indicated its ability and desire to process a bw-call-completion request (in the 6xx response), SUBSCRIBE requests should typically not fail. However, in some conditions, a terminating server might indicate availability of the bw-call-completion package in the 6xx response, but reject the SUBSCRIBE request because the maximum number of terminator sessions is reached. In this case, a 480 Temporarily Unavailable response is sent from the terminating server to the originating server.

3.42.1.3 bw-call-completion NOTIFY Messages

The SUBSCRIBE and NOTIFY exchange follows *RFC 3265* and uses the “bw-call-completion” event package name. The attributes supported for NOTIFY message bodies are slightly different from the attributes in the SUBSCRIBE bodies. Cisco BroadWorks only supports two attributes for NOTIFY events.

- The first supported attribute is *queue-type* and is used in a similar manner as the SUBSCRIBE messages. Since Cisco BroadWorks only supports CCBS in this release, this value is always used.
- The second supported attribute is the *queue-state* attribute. This attribute has two supported values: “request-queued” and “user-free-for-recall”. The “request-queued” value indicates to the originator that the terminator is currently busy and that callback should not be tried at this time. This is typically used in the initial NOTIFY message; however, it can also be used when the terminator becomes busy once again during the callback phase (this can occur in some cases if the recall setup takes some time). In this case, recall should be aborted on the originator side. The “user-free-for-recall”, as its name implies, indicates to the originator that the terminator is now available to receive a callback.

The *Subscription-state* header is typically set to “active”; however, it is also set to “terminated” in the NOTIFY sent after a terminating SUBSCRIBE is received. The *expires* parameter is set to the current remaining value of the subscription when the NOTIFY message is sent.

Following is an example of a SIP NOTIFY message sent from the terminating Application Server to indicate that a request was received and queued.

```
NOTIFY sip:as1.example1.org:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.8.24;branch=z9hG4bKBroadWorks.1jmom3n-
;event=bw-call-completion
From: <sip: +15146959662@as2.example2.org;user=phone>;tag=0987654321-
```

```
To: <sip: +15146993604@as1.example1.org;user=phone>;tag=1234567890-
Call-ID: BW123456789012345678901234@192.168.8.24
Cseq: 114594655 NOTIFY
Contact:<sip:as2.example1.org:5060>
Max-Forwards: 70
Event: bw-call-completion;id=145
Subscription-state: active;expires=107990
Content-Type: application/bw-call-completion
Content-Length: 45

queue-nature: ccbs
queue-state: request-queued
```

Following is an example of a SIP NOTIFY message sent from the terminating Application Server to indicate that the terminator is ready for recall.

```
NOTIFY sip:as1.example1.org:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.8.24;branch=z9hG4bKBroadWorks.1jmom3n-
;event=bw-call-completion
From: <sip: +15146959662@as2.example2.org;user=phone>;tag=0987654321-
To: <sip:5146993604@as1.example1.org;user=phone>;tag=1234567890-
Call-ID: BW123456789012345678901234@192.168.8.24
Cseq: 114594701 NOTIFY
Contact: <sip:as2.example1.org:5060>
Max-Forwards: 70
Event: bw-call-completion;id=145
Subscription-state: active;expires=63440
Content-Type: application/bw-call-completion
Content-Length: 52

queue-nature: ccbs
queue-state: user-free-for-recall
```

Following is an example of a SIP NOTIFY message sent from the terminating Application Server to indicate subscription termination.

```
NOTIFY sip:as1.example1.org:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.8.24;branch=z9hG4bKBroadWorks.1jmom3n-
;event=bw-call-completion
From: <sip: +15146959662@as2.example2.org;user=phone>;tag=0987654321-
To: <sip:+15146993604@as1.example1.org;user=phone>;tag=1234567890-
Call-ID: BW123456789012345678901234@192.168.8.24
Cseq: 114594888 NOTIFY
Contact: <sip:as2.example1.org:5060>
Max-Forwards: 70
Event: bw-call-completion;id=145
Subscription-state: terminated
Content-Length: 0
```

3.42.1.4 Syntax

SIP responses can contain an *Allow-Events* header with the bw-call-completion value. Typically, this header would be included in SIP 600 or 606 responses; however, it can also be included in different SIP responses depending on configurable treatments.

```
Allow-Events = ( "Allow-Events" / "u" ) HCOLON event-type *
               (COMMA event-type)

event-type = "bw-call-completion" / token
```

The *Event* header can contain the bw-call-completion value in SIP SUBSCRIBE and NOTIFY requests. In addition, the *Event* header for the bw-call-completion event package contains the *id* parameter to allow easy correlation of SUBSCRIBE and NOTIFY requests.

```
Event =  ( "Event" / "o" ) HCOLON event-type *( SEMI event-param )
event-type  = "bw-call-completion" / token
event-param =  ( "id" EQUAL token )
```

The *Accept* header can contain the application/bw-call-completion value in SIP SUBSCRIBE requests. This value is implicitly added for all SUBSCRIBE requests in which the *Event* header contains the bw-call-completion value.

```
Accept =  ( "Accept" ) HCOLON media-range
media-range = "application/bw-call-completion" / token
```

The *Content-Type* header can contain the application/bw-call-completion value in SIP SUBSCRIBE and NOTIFY requests.

```
Content-Type =  ( "Content-Type" / "c" ) HCOLON media-type
media-type = "application/bw-call-completion"
```

The *Via* header can contain an *event* parameter with bw-call-completion value in SUBSCRIBE and NOTIFY requests as well as in the corresponding responses.

```
Via         =  ( "Via" / "v" ) HCOLON via-parm *(COMMA via-parm)
via-parm    =  sent-protocol LWS sent-by *( SEMI via-params )
via-params  =  via-event
via-event   =  "event" EQUAL ("bw-call-completion" / token)
```

The SIP SUBSCRIBE and NOTIFY messages for the bw-call-completion event-package carry message bodies of type "application/bw-call-completion". This body type is formally defined as follows.

```
Bw-call-completion = [queue-nature CLRF][queue-operation CLRF]
                     [queue-state CLRF][caller CLRF]
                     [service-retention CLRF]
queue-nature        =  "queue-nature" HCOLON "CCBS"
queue-operation     =  "queue-operation" HCOLON ( "add" /
                       "suspend" / "resume" )
queue-state         =  "queue-state" HCOLON ( "request-queued" /
                       "user-free-for-recall" )
caller              =  "caller" HCOLON addr-spec
service-retention   =  "service-retention"
```

3.42.2 Legacy Automatic Callback

Legacy Automatic Callback (ACB) allows a subscriber to reach another busy subscriber shortly after that busy subscriber becomes available. When active, the service monitors the busy subscriber. When that subscriber is available, it recalls the original caller, places a call back to the original busy subscriber, and connects the two parties.

Though similar to Call Completion Services in purpose, the Legacy Automatic Callback service is different in important ways. By design, it interworks with the same service implemented in the Nortel CS 2000 softswitch. Cisco BroadWorks implements both the originating and terminating switch behavior. Thus, a subscriber hosted on Cisco BroadWorks can use the Automatic Callback service to reach a subscriber hosted on a Nortel CS 2000 softswitch, and vice versa. A caller on Cisco BroadWorks may also use Automatic Callback to reach another subscriber hosted on Cisco BroadWorks.

The implementation also involves a GENBAND C3 softswitch, which facilitates the interworking between the SIP signaling used in Cisco BroadWorks and the SS7 signaling used in the Nortel CS 2000. *Figure 64* illustrates the required architecture. On the Nortel CS 2000 side, the GENBAND C3 provides an SS7 interface, where it sends and receives TCAP messages to and from the Nortel CS 2000. On the Cisco BroadWorks side, the GENBAND C3 provides a SIP interface based on the SIP event notification framework. The GENBAND C3 and Cisco BroadWorks exchange SIP SUBSCRIBE and NOTIFY requests and the corresponding responses. These Session Interface Protocol (SIP) messages carry eXtensible Markup Language (XML) message bodies, which encode the same data that is contained in the Transactional Capabilities Application Part (TCAP) messages.

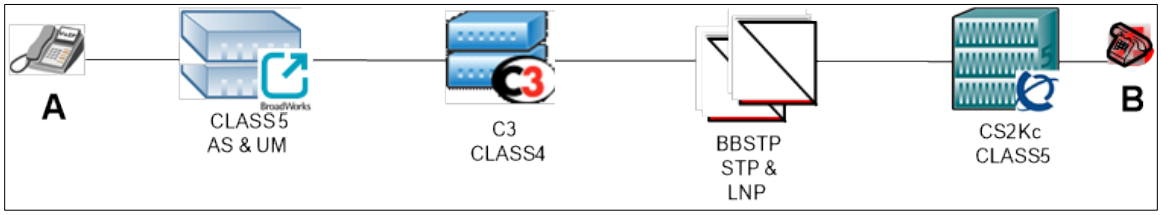

Figure 64 Legacy Automatic Callback Architecture Diagram

3.42.2.1 Origination

This section show the signaling flow when the originating user is hosted on Cisco BroadWorks application server.

The Automatic Callback request activation procedure begins when the calling party, a Cisco BroadWorks user, dials the Automatic Callback FAC. *Figure 65* illustrates the steps that the Application Server executes upon detecting the FAC.

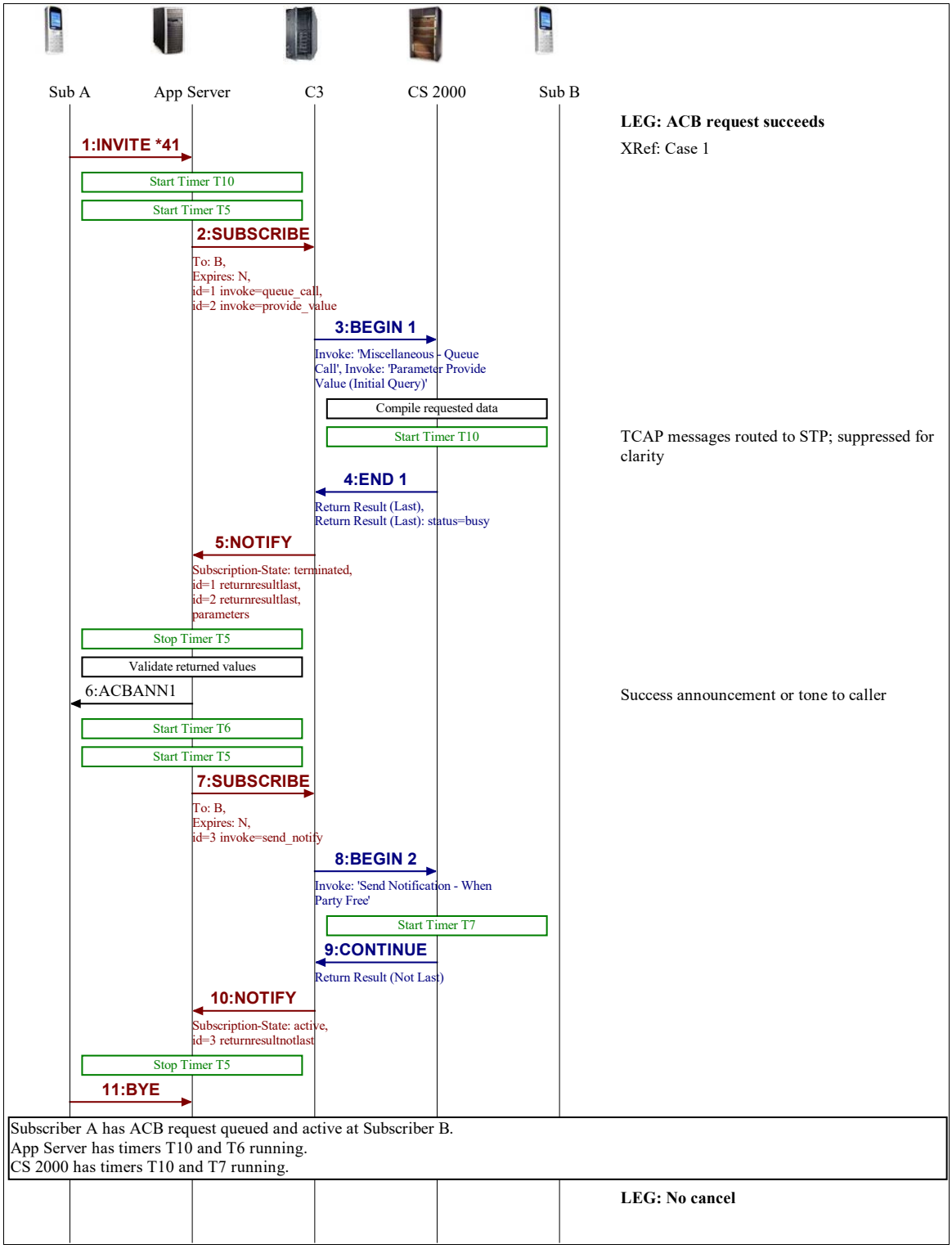

Figure 65 Cisco BroadWorks to NT: Automatic Callback Request Activation Succeeds

1) Subscriber A dials the Automatic Callback FAC, shown as *41 in the diagram. The Application Server starts Timer T10 which is a protective timer to ensure that the Automatic Callback request cannot become a leaked resource.

   The Application Server also starts Timer T5 which defines the maximum amount of time the Application Server will wait for the NOTIFY request from the other SIP User Agent (UA).

2) The Application Server sends a SUBSCRIBE request to the GENBAND C3.

   The SUBSCRIBE request contains an XML message body that corresponds to the TCAP BEGIN 1 message. This message asks the terminating switch to create an Automatic Callback queue entry against the called party and to return status information about the called party.

   The Application Server sends the SUBSCRIBE request to the network and the normal network-side routing causes the request to be delivered the GENBAND C3. This routing is not shown.

3) The GENBAND C3 sends a BEGIN 1 message to the CS 2000.

4) The CS 2000 sends an END 1 message to the GENBAND C3.

5) The GENBAND C3 sends a NOTIFY request to the Application Server.

   The NOTIFY request contains an XML message body corresponding to the TCAP END 1 message.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the completion of the TCAP transaction, the GENBAND C3 terminates the subscription and places a *Subscription-State* header with the value “terminated” in the NOTIFY request.

   When the Application Server receives the NOTIFY request, it stops the Timer T5 and validates the data in the XML message body.

   This scenario assumes that data validation passes.

6) The Application Server plays a success announcement to the calling party.

   Because the terminating switch indicated that the Automatic Callback queue request was granted, the Application Server starts the Timer T6, which defines the amount of time that the Automatic Callback request is allowed to be valid. If the Timer T6 expires, the Application Server automatically cancels the Automatic Callback request.

   At this point, the terminating switch has an Automatic Callback entry in its queue, but the entry is not yet active.

7) The Application Server sends a SUBSCRIBE request to the GENBAND C3.

   This SUBSCRIBE request contains an XML message body that corresponds to the TCAP BEGIN 2 message. This message requests the terminating switch to a send a notification when the called party is idle, thus making the Automatic Callback queue request active. The message also contains a value for Timer T7 to be used by the terminating switch.

8) The GENBAND C3 sends a BEGIN 2 to the CS 2000.

   When the CS 2000 receives the BEGIN 2 message, it activates the Automatic Callback queue request and starts Timer T7. The CS 2000 sets the Timer T7 value from the value it received in the BEGIN 2 message.

9) The CS 2000 sends a CONTINUE message to the GENBAND C3.

10) The GENBAND C3 sends a NOTIFY request to the Application Server.

The NOTIFY request contains an XML body that corresponds to the TCAP CONTINUE message.

This NOTIFY request is a reply to the earlier SUBSCRIBE request. Because the TCAP transaction is not completed, the GENBAND C3 keeps the subscription active and places a Subscription-State header with the value “active” in the NOTIFY request.

At this point, the CS 2000 has an Automatic Callback request queued and active. The Application Server also has resources allocated for Automatic Callback.

After an Automatic Callback request is queued at the CS 2000, it waits for the called party to become idle. When this happens, it waits a short time controlled by the guard timer and then it triggers a series of steps that cause the Application Server to recall the calling party.

Figure 66 illustrates the scenario where the calling party answers the recall is pictured in the flow diagram.

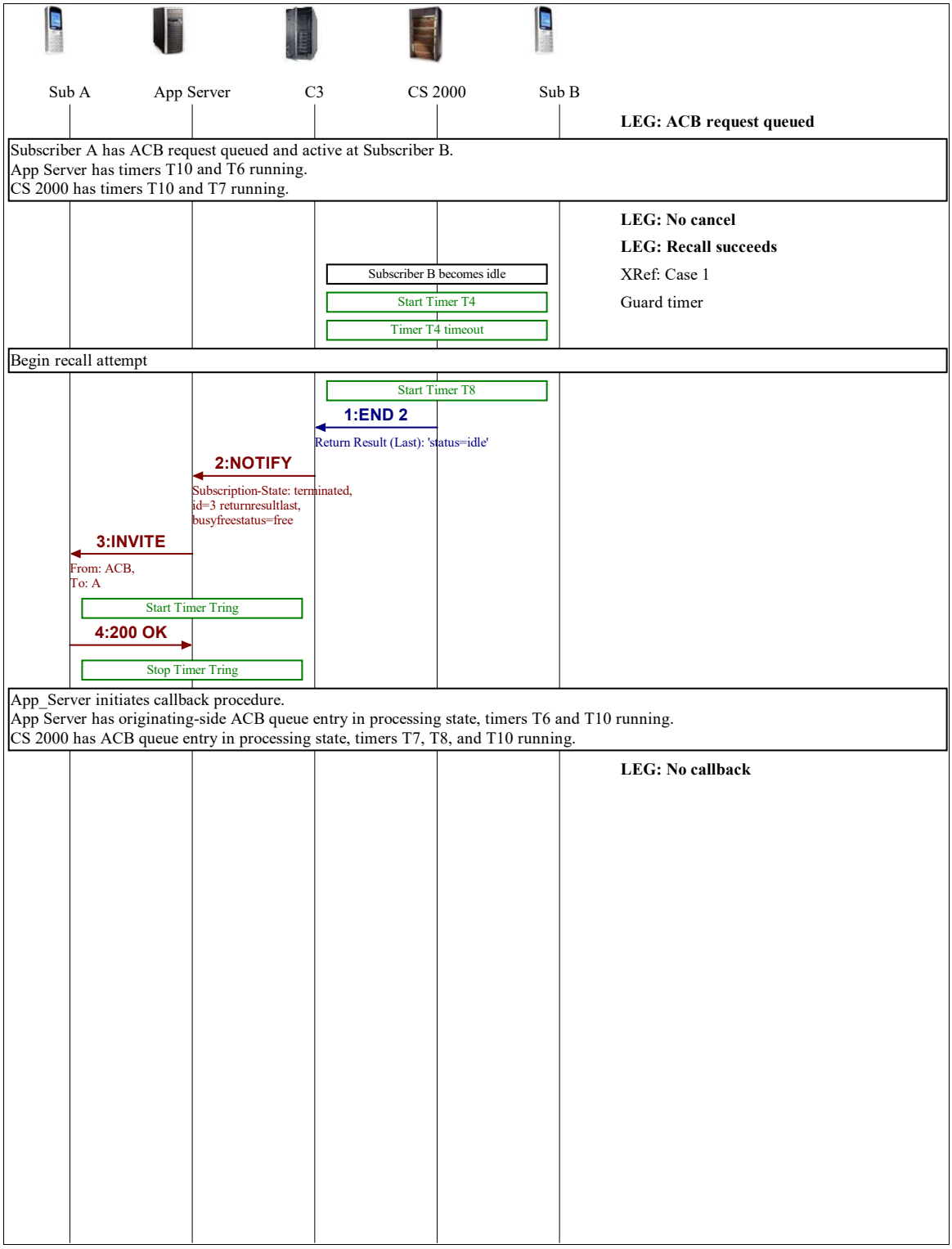

Figure 66 Cisco BroadWorks to NT: Recall Succeeds

The scenario begins when the CS 2000 detects that the called party has become idle. This event causes the CS 2000 to start the guard timer, which allows the called party to make or answer another call before the CS 2000 notifies the originating switch.

In this scenario, the guard timer expires, and the CS 2000 begins the steps that cause a recall to the calling party.

In the first step, the CS 2000 starts Timer T8. This timer controls the length of time allowed for the originating switch to respond to the idle status notification. If Timer T8 expires, and if there are other Automatic Callback requests in the queue, then the CS 2000 may make the current queue entry inactive and send a notification for the next active request.

1) The CS 2000 sends an END 2 message to the GENBAND C3.
   The END 2 message indicates that the status of the called party is idle.

2) The GENBAND C3 sends a NOTIFY request to the Application Server.

   The NOTIFY request contains an XML message body corresponding to the TCAP END 2 message.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the completion of the TCAP transaction, the GENBAND C3 terminates the subscription and places a *Subscription-State* header with the value “terminated” in the NOTIFY request.

   When the Application Server receives the NOTIFY request, it decodes the XML message body and discovers that the called party is available. This triggers the recall to the calling party.

   Before initiating the recall, the Application Server starts Timer Tring. This timer limits the amount of time allowed for the calling party to answer the recall.

3) The Application Server sends an INVITE request to Subscriber A’s device.

4) Subscriber A’s device sends a *200* response to the Application Server.

When the Application Server receives the *200* response, it stops Timer Tring and begins the steps to execute the callback to the called party.

After the calling party answers the recall, the Application Server begins a series of steps to place a call back to the called party.

The first step is for the Application Server to query the terminating switch to verify that the called party is still idle. If the query response indicates that the called party is still idle, then the Application Server assumes a callback will succeed and then frees all Automatic Callback resources and continues with a normal origination from subscriber A to subscriber B. Otherwise, if the query response indicates that subscriber B has become busy again, then the Application Server plays an appropriate announcement to subscriber A and reactivates the Automatic Callback queue request.

Figure 67 illustrates the scenario where the called party is available after the calling party answers the recall.

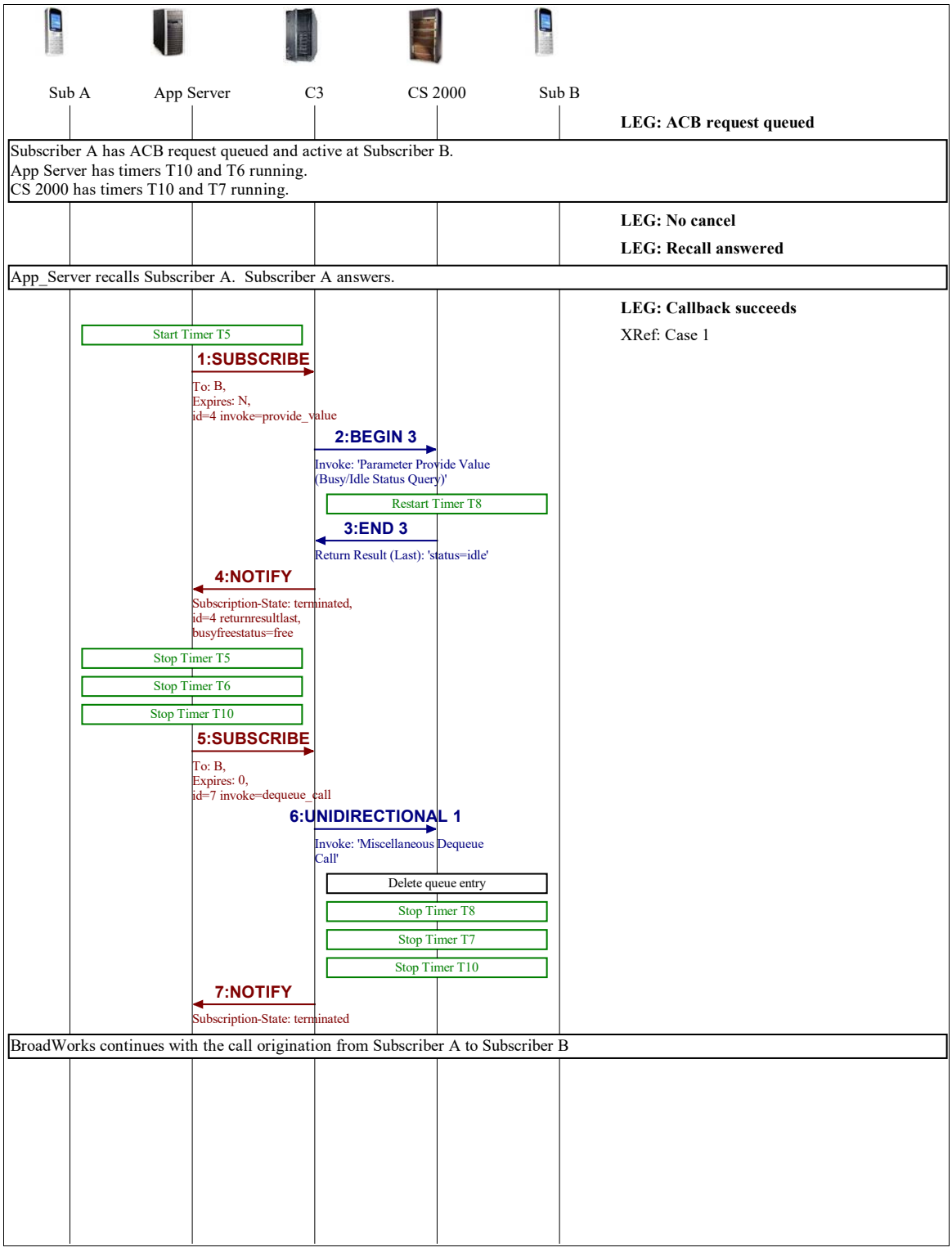

Figure 67 Cisco BroadWorks to NT: Callback Proceeds

This scenario begins when the calling party answers the recall.

1) The Application Server sends a SUBSCRIBE request to the GENBAND C3.
The SUBSCRIBE request contains an XML message body that corresponds to the TCAP BEGIN 3 message. This message requests the terminating switch to return the busy/idle status information about the called party.

2) The GENBAND C3 sends a BEGIN 3 message to the CS 2000.

   When the CS 2000 receives the BEGIN 3 message, it checks the busy/idle status of the called party. In this scenario the called party, subscriber B, is idle.

   Before sending the status notification, the CS 2000 sets Timer 8.

3) The CS 2000 sends an END 3 message to the GENBAND C3.

   The END 3 message indicates that the status of the called party is idle.

4) The GENBAND C3 sends a NOTIFY request to the Application Server.

   The NOTIFY request contains XML message body corresponding to the TCAP END 3 message.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the completion of the TCAP transaction, the GENBAND C3 terminates the subscription and places a *Subscription-State* header with the value “terminated” in the NOTIFY request.

   When it receives the NOTIFY request, the Application Server decodes the XML message body and examines the data it contains. In this case, the data indicates that subscriber B is still available. In response, the Application Server stops all timers and begins the steps to delete the Automatic Callback queue request at the terminating switch.

5) The Application Server sends a SUBSCRIBE request to the GENBAND C3.

   The SUBSCRIBE request contains an XML message body that corresponds to the TCAP UNIDIRECTIONAL 1 message. This message requests the terminating switch to delete Automatic Callback queue entry.

   Because the UNIDIRECTIONAL 1 message does not begin a new TCAP transaction, the subscription will not persist. For this reason, the Application Server places an *Expires* header with the value “0” in the SUBSCRIBE request.

6) The GENBAND C3 sends a UNIDIRECTIONAL 1 message to the CS 2000.

   When the CS 2000 receives the TCAP message, it deletes the Automatic Callback queue entry and frees all resources related to this Automatic Callback queue request.

7) The GENBAND C3 sends a NOTIFY request to the Application Server.

   The NOTIFY request does not contain a message body.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the Expires header in the SUBSCRIBE request, the GENBAND C3 terminates the subscription and places a *Subscription-State* header with the value “terminated” in the NOTIFY request.

   At this point, the Application Server frees all Automatic Callback-related resources for this instance and sends an INVITE request to the network to initiate a call from subscriber A to subscriber B.

In the rare case that subscriber B becomes busy once again between the time the CS 2000 sends the END 3 message and the time the CS 2000 receives the IAM message for the termination, subscriber A would receive busy treatment. In this case, the Application Server does not automatically reactivate Automatic Callback. Subscriber A, however, may choose to reactivate Automatic Callback.

3.42.2.2 Termination

As mentioned previously, Cisco BroadWorks can take the role of either the originating switch or the terminating switch. This section describes the details of Cisco BroadWorks acting as the terminating switch.

The Automatic Callback request activation procedure begins when the calling party, a user hosted on the CS 2000, dials the Automatic Callback FAC. Upon detecting the FAC, the CS 2000 executes the steps to create and activate an Automatic Callback queue request at the Application Server.

Figure 68 illustrates the scenario where the Automatic Callback request is successfully queued.

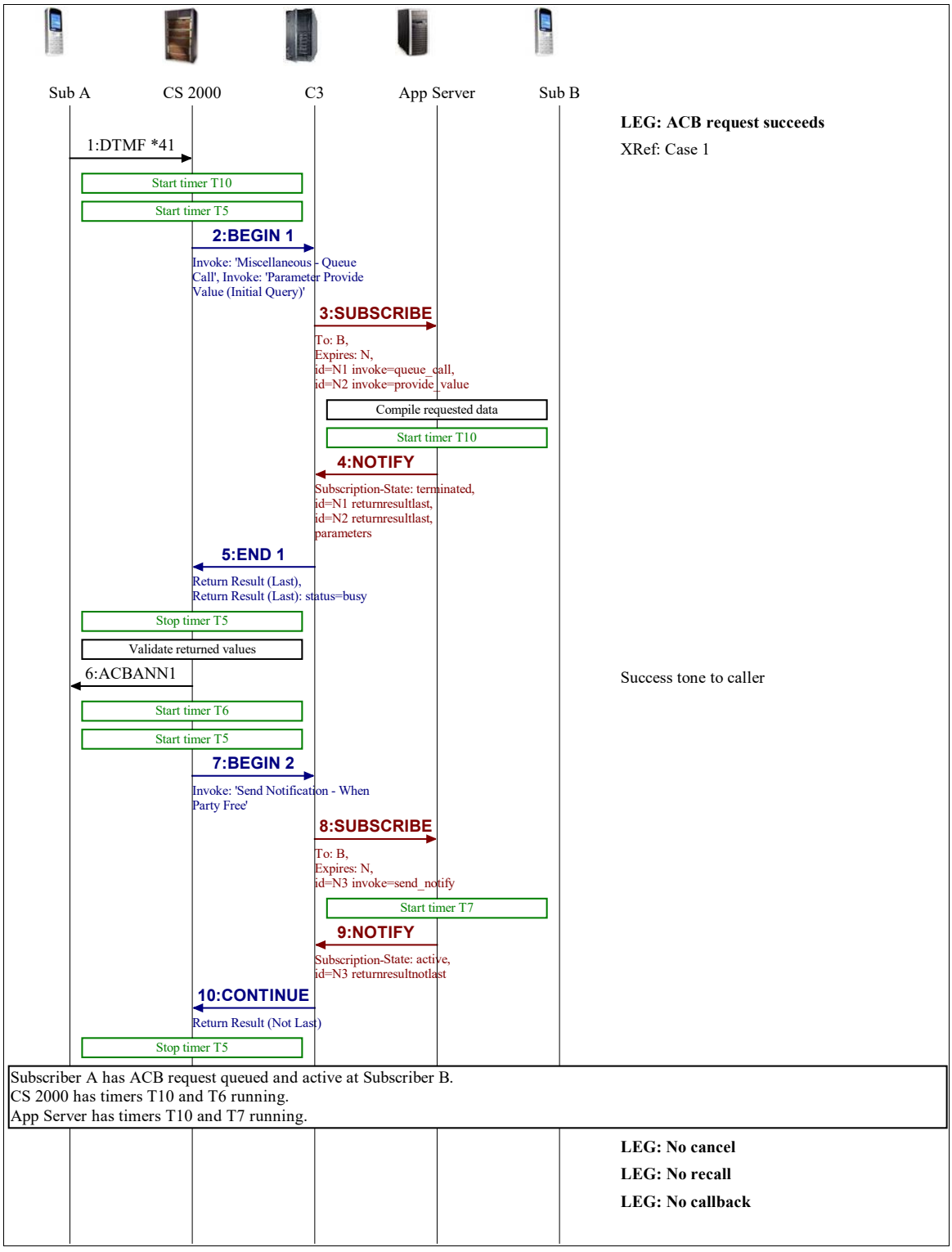

Figure 68 NT to Cisco BroadWorks: Successful Automatic Callback Queue Request

1) Subscriber A dials the Automatic Callback FAC, shown as *41 in the diagram.
2) The CS 2000 sends a BEGIN 1 message to the GENBAND C3.

   The BEGIN 1 message requests the terminating switch to create an Automatic Callback queue entry against the called party and to return status information about the called party.

3) The GENBAND C3 sends a SUBSCRIBE request to the Application Server.

   The SUBSCRIBE request contains an XML message body that corresponds to the TCAP BEGIN 1 message.

   When the Application Server receives the SUBSCRIBE request, it takes these actions:

   – Gather information about the service profile of subscriber B.

   – Create an Automatic Callback queue entry for subscriber A against subscriber B. This entry is marked as inactive.

   – Start Timer T10.

4) The Application Server sends a NOTIFY request to the GENBAND C3.

   The NOTIFY request contains an XML message body that corresponds to the TCAP END 1 message.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the completion of the TCAP transaction, the Application Server terminates the subscription and places a Subscription-State header with the value “terminated” in the NOTIFY request.

5) The CS 2000 sends an END 1 message to the GENBAND C3.

   The CS 2000 validates the data in the END 1 message. This scenario assumes the data validation passes.

6) The CS 2000 plays a success announcement to subscriber A.

7) The CS 2000 sends a BEGIN 2 message to the GENBAND C3.

   The BEGIN 2 message requests the terminating switch to a send a notification when the called party is idle, thus making the Automatic Callback queue request active. The message also contains a value for Timer T7 to be used by the terminating switch.

8) The GENBAND C3 sends a SUBSCRIBE request to the Application Server.

   This SUBSCRIBE request contains an XML message body that corresponds to the TCAP BEGIN 2 message.

   When the Application Server receives the SUBSCRIBE request, it activates the Automatic Callback queue request and starts Timer T7. The Application Server sets the Timer T7 value from the value it received in the XML message body of the SUBSCRIBE request.

9) The Application Server sends a NOTIFY request to the GENBAND C3.

   The NOTIFY request contains an XML body that corresponds to the TCAP CONTINUE message.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. Because the TCAP transaction will not be completed, the Application Server keeps the subscription active and places a Subscription-State header with the value “active” in the NOTIFY request.

10) The GENBAND C3 sends a CONTINUE message to the CS 2000.

    At this point, the Application Server has an Automatic Callback request queued and active.

After an Automatic Callback request is queued at the Application Server, it waits for the called party to become idle. When this happens, it triggers a series of steps that cause the originating switch to recall the calling party.

Figure 69 illustrates the scenario where the calling party answers the recall.

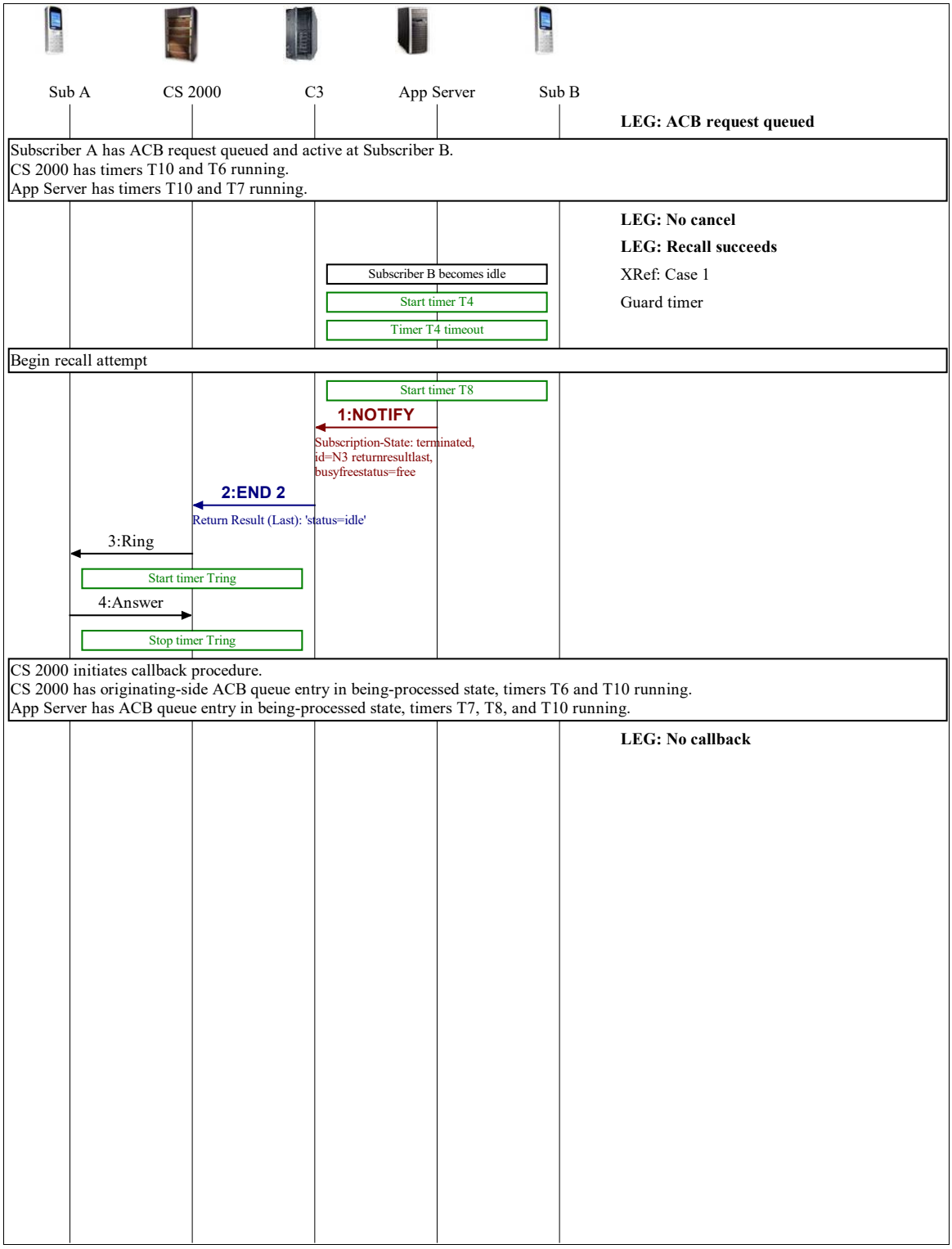

Figure 69 NT to Cisco BroadWorks: Recall Succeeds

The scenario begins when the Application Server detects that the called party has become available. This event causes the Application Server to start the guard timer, which allows the called party to make or answer another call before the Application Server notifies the originating switch.

In this scenario, the guard timer expires, and the Application Server begins the steps that cause a recall to the calling party.

In the first step, the Application Server starts Timer T8. This timer controls the length of time allowed for the originating switch to respond to the idle status notification. If Timer T8 expires, and if there are other Automatic Callback requests in the queue, then the Application Server may make the current queue entry inactive, and send a notification for the next active request.

1) The Application Server sends a NOTIFY request to the GENBAND C3.
   The NOTIFY request also contains an XML message body corresponding to the TCAP END 2 message. This message notifies the originating switch that the called party's status is idle.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the completion of the TCAP transaction, the Application Server terminates the subscription and places a *Subscription-State* header with the value "terminated" in the NOTIFY request.

2) The GENBAND C3 sends an END 2 message to the CS 2000.

3) The CS 2000 alerts Subscriber A.

4) Subscriber A answers.

   When the subscriber A answers, the CS 2000 stops Timer Tring and begins the steps to execute the callback to the called party.

Figure 70 illustrates the scenario where the called party is available after the calling party answers the recall.

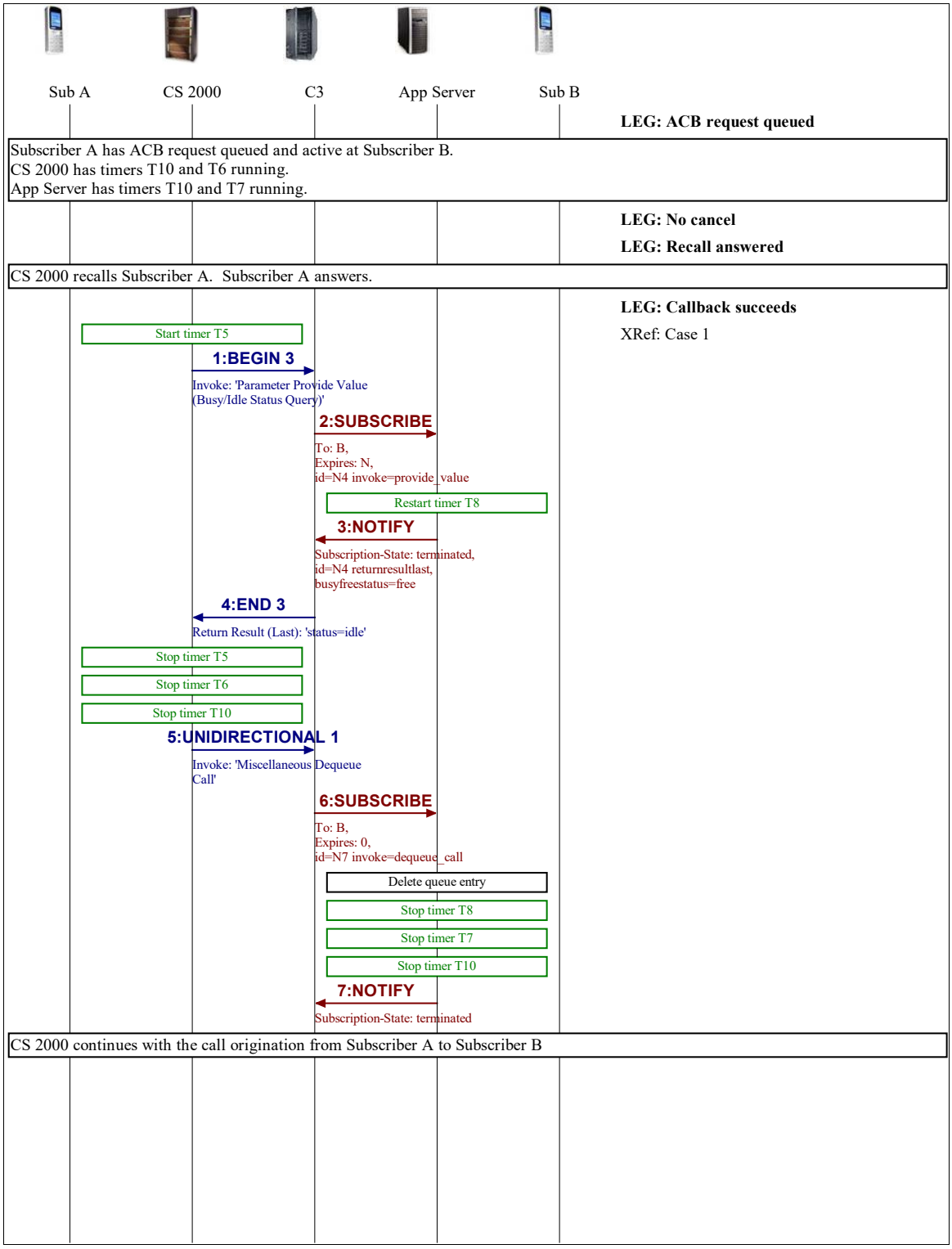

Figure 70 NT to Cisco BroadWorks: Callback Proceeds

This scenario begins when the calling party answers the recall.

1) The CS 2000 sends a BEGIN 3 message to the GENBAND C3.
The BEGIN 3 message requests the terminating switch to return the busy/idle status information about the called party.

2) The GENBAND C3 sends a SUBSCRIBE request to the Application Server.

   The SUBSCRIBE request contains an XML message body that corresponds to the TCAP BEGIN 3 message.

3) The Application Server sends a NOTIFY request to the GENBAND C3.

   The NOTIFY request contains an XML message body that corresponds to the TCAP END 3 message. This message indicates that the called party's status is Idle.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the completion of the TCAP transaction, the GENBAND C3 terminates the subscription and places a *Subscription-State* header with the value "terminated" in the NOTIFY request.

4) The GENBAND C3 sends an END 3 message to the CS 2000.

5) The CS 2000 sends a UNIDIRECTIONAL 1 message to the GENBAND C3.

   The UNIDIRECTIONAL 1 message requests the terminating switch to delete Automatic Callback queue entry.

6) The GENBAND C3 sends a SUBSCRIBE request to the Application Server.

   The SUBSCRIBE request contains an XML message body that corresponds to the TCAP UNIDIRECTIONAL 1 message.

   Because the UNIDIRECTIONAL 1 message does not begin a new TCAP transaction, the subscription will not persist. For this reason, the GENBAND C3 places an Expires header with the value "0" in the SUBSCRIBE request.

7) The Application Server sends a NOTIFY request to the GENBAND C3.

   The NOTIFY request does not contain a message body.

   This NOTIFY request is a reply to the earlier SUBSCRIBE request. In accordance with the *Expires* header in the SUBSCRIBE request, the Application Server terminates the subscription and places a *Subscription-State* header with the value "terminated" in the NOTIFY request.

   At this point, the CS 2000 frees all Automatic Callback-related resources for this instance and initiates a new call from subscriber A to subscriber B.

3.42.2.3 Content-Type application/acb+xml

Content-Type "application/acb+xml" is defined for inclusion within SUBSCRIBE and NOTIFY to support Legacy Automatic Callback. The application/acb+xml bodies are implemented using standard XML as defined in the following XML schema.

The application/acb+xml schema is as follows.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="acbtcap" type="acbtcapType">
    <xs:annotation>
      <xs:documentation>
```

```
        Content-Type "application/acb+xml" is defined for inclusion
        within SIP SUBSCRIBE and NOTIFY requests for event package
        legacy-acb to support Legacy Automatic Callback.  The element
        and attribute definitions within the schema closely follow the
        "ACB PROTOCOL, TECHNICAL REQUIREMENTS, TTS, ACB Protocol Spec,
        710 00158 AAAA-DS Ed. 0 19/7/98".  The top level element
        "acbtcap" aligns to the component section of the TCAP message
        and can carry multiple TCAP components.  Inside the "acbtcap"
        element is the "component" element which contains an invoke,
        return result, return error, or reject components.  Within these
        components are the various parameters passed between the two
        platforms.

        Although not specifically indicated within this schema, the
        encoding must be UTF-8.

        Although not all the int ranges within this schema are clearly
        specified, the int values should be >= 0 unless otherwise
        indicated.
      </xs:documentation>
    </xs:annotation>
</xs:element>

<xs:complexType name="acbtcapType">
    <xs:annotation>
      <xs:documentation>
       The components are described within the componentType. The
       transaction_portion_error indicates that the transaction was
       malformed (such as by not having an invoke ID) and a better
       failure response could not be provided.  The
       transaction_route_failure indicates denial due to offnet user and
       a better failure response could not be provided.  The
       transaction_timeout indicates that the transaction expired and a
       better failure response could not be provided.  If it occurs to
       mid ACB transaction (such as for send_notify or provide_value),
       the receiver may handle the failure however desired; however it
       should assume call still queued.  If transaction_timeout received
       for 199equeuer or cancel request, the receiver is not required to
       attempt delivery of another 199equeuer or cancel.
      </xs:documentation>
    </xs:annotation>
    <xs:choice>
      <xs:element name="component" type="componentType"
                  minOccurs="0" maxOccurs="2"/>
      <xs:element name="transaction_portion_error"
                  type="xs:string"/>
      <xs:element name="transaction_route_failure"
                  type="xs:string"/>
      <xs:element name="transaction_timeout"
                  type="xs:string"/>
    </xs:choice>
</xs:complexType>

<xs:complexType name="componentType">
    <xs:annotation>
      <xs:documentation>
        The invoke component is a request to do something; the
        returnresultlast, returnresultnotlast, returnerror, and reject
        components are the potential corresponding responses related to
        the invoke component. The returnresultlast, returnerror, and
        reject are final responses to invoke; returnresultnotlast is an
        early (non final) response to invoke.
```

```
      The invoke components are sent within SUBSCRIBE; the related
      response components are sent within NOTIFY.  Response components
      (i.e. returnresultlast, returnresultnotlast, returnerror, and
      reject) should not be sent within a SUBSCRIBE or SIP responses
      since they are considered unexpected and may be ignored.
      Although SIP error responses can indirectly reject an invoke, it
      is recommended (not required) to indicate such failures within
      the NOTIFY's components instead of returning a SUBSCRIBE failure
      response when both adequately communicate the failure reason.

      The component ID in the response component is equivalent to the
      component ID within the invoke component.  Component IDs have a
      range inclusively between 0 and 127; they are only unique to SIP
      dialog and TCAP transaction.  After the transaction completes,
      the ID may be reused.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element name="invoke" type="invokeType"/>
    <xs:element name="returnresultlast"
                type="returnresultlastType"/>
    <xs:element name="returnresultnotlast"
                type="returnresultnotlastType"/>
    <xs:element name="returnerror" type="returnerrorType"/>
    <xs:element name="reject" type="rejectType"/>
  </xs:choice>
  <xs:attribute name="id" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="invokeType">
  <xs:annotation>
    <xs:documentation>
     Invoking queue_call is a request to queue call for ACB.  Invoking
     provide_value is a request to obtain data related to
     appropriateness to queue (or continue to queue) call for ACB.
     Invoking send_notify is a request to be informed when called party
     is free.  Invoking 200equeuer_call is a request to de-queue a call
     queued for ACB.  Invoking cancel is a request to cancel a non
     complete invoke request.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="parameters" type="parametersType"/>
  </xs:sequence>
  <xs:attribute name="opcode" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="queue_call"/>
        <xs:enumeration value="provide_value"/>
        <xs:enumeration value="send_notify"/>
        <xs:enumeration value="200equeuer_call"/>
        <xs:enumeration value="cancel"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="returnresultlastType">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="parameters" type="parametersType"/>
  </xs:sequence>
```

```
</xs:complexType>

<xs:complexType name="returnresultnotlastType">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="parameters" type="parametersType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="returnerrorType">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="parameters" type="parametersType"/>
  </xs:sequence>
  <xs:attribute name="code" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="task_refused"/>
        <xs:enumeration value="queue_full"/>
        <xs:enumeration value="no_queue"/>
        <xs:enumeration value="unexpected_data_value"/>
        <xs:enumeration value="not_queued"/>
        <xs:enumeration value="unavailable_resource"/>
        <xs:enumeration value="timer_expired"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="rejectType">
  <xs:annotation>
    <xs:documentation>
      The reject component contains a type (int of range 0-3) and
      specifier (int of range 0-7).  The values correspond to section
      B.4.21 of the mentioned "ACB PROTOCOL, TECHNICAL REQUIREMENTS"
      specification except that the type range within acb+xml starts
      with 0 instead of 128.

      The following are the problem types: 0) General, 1) Invoke, 2)
      Return Result, and 3) Return Error.  Each type has a
      corresponding list of specifiers.

      The following are General specifiers: 0) Unrecognized component,
      1) Mistyped component, and 2) Badly structured component.

      The following are Invoke specifiers: 0) Duplicate invoke ID, 1)
      Unrecognized operation, 2) Mistyped parameter, 3) Resource
      limitation, 4) Initiating release, 5) Unrecognized linked ID, 6)
      Linked response unexpected, and 7) Unexpected linked operation.

      The following are Return Result specifiers: 0) Unrecognized
      invoke   ID, 1) Return result unexpected, and 2) Mistyped
      parameter.

      The following are Return Error specifiers: 0) Unrecognized invoke
      ID, 1) Return error unexpected, 2) Unrecognized error, 3)
      Unexpected error, and 4) Mistyped parameter.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="parameters" type="parametersType"/>
  </xs:sequence>
  <xs:attribute name="type" use="required" type="xs:int"/>
  <xs:attribute name="specifier" use="required"
```

```
                type="xs:int"/>
</xs:complexType>

<xs:complexType name="parametersType">
  <xs:annotation>
    <xs:documentation>
      All of the components may contain the following parameters:
      busyfreestatus, callfwdactive, termrestrict, dntoline, duration,
      operationid, and digits.  However if not actually expected, the
      parameters may be ignored.  Within an invoke, the following
      parameters contain no actual value since requesting the value:
      busyfreestatus, callfwdactive, termrestrict, and dntoline.

      The details of parameters busyfreestatus, callfwdactive,
      termrestrict, dntoline, and digits are presented within their
      corresponding types.  The operationid is the related component
      ID.  For instance when included within an invoke cancel, the
      operationid is the component ID of invoke that is being canceled.

      The duration is the seconds requested for the associated invoke.
      For instance when included within an invoke send_notify
      (currently the only expected usage), the duration indicates the
      amount of time the invoker is willing to wait for notification of
      the state change.  The duration cannot be extended by re-
      subscribes. To avoid refreshing SIP subscriptions, it is
      recommended that the SIP subscription's expiration interval be at
      least as high as this duration.  The subscription target should
      not increase the received duration (except for the variations
      caused by message delivery and processing).  The target can lower
      the duration; however it should not indicate the adjustment
      within the related NOTIFY sent during activation.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="busyfreestatus" type="busyfreestatusType"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="callfwdactive" type="callfwdactiveType"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="termrestrict" type="termrestrictType"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="dntoline" type="dntolineType"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="duration" type="xs:int"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="operationid" type="xs:int"
                minOccurs="0" maxOccurs="1"/>
    <xs:element name="digits" type="digitsType"
                minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="digitsType">
  <xs:annotation>
    <xs:documentation>
```

```
        The digit is a string of type calling or called.  The string
        would typically correlate to the sip-uri contained within the
        SUBSCRIBE's request-uri or From.  The digits string should
        contain digits (such as RFC 3966 global-number-digits or local-
        number-digits) from the user portion of sip-uri; however the
        full uri may also be used.  An empty string for digits is valid.
        The empty string may be (not required to be) used when the sip-
        uri contained no user portion; however it is not limited to such
        a usage.

        The digits are considered unexpected within invoke since the
        called and calling parties are determined respectively by the
        SUBSCRIBE's or subscription's) request-uri and From.
      </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="called"/>
              <xs:enumeration value="calling"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="busyfreestatusType">
    <xs:annotation>
      <xs:documentation>
        The busyfreestatus indicates the state of invoked party: busy or
        free.  If the invoked party is busy but able to receive
        additional calls (for instance by using Call Waiting service),
        free may be indicated as the busyfreestatus.  If the invoke
        party is currently not registered, busy may be indicated as the
        busyfreestatus.  The empty string is for use within invoke to
        request busyfreestatus.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value=""/>
      <xs:enumeration value="busy"/>
      <xs:enumeration value="free"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="callfwdactiveType">
    <xs:annotation>
      <xs:documentation>
```

```
        The callfwdactive indicates the forwarding status (or request
        for status) of called (invoked) party concerning cfu (call
        forwarding unconditional or call forwarding always), cfbl (call
        forwarding busy line or call forwarding busy), cfda (call
        forwarding don't answer or call forwarding no answer), and scf
        (selective call forwarding).  The following are the potential
        status values: notsupported, active, inactive, and spare.
        Unless spare is known to reflect otherwise, it should not be
        interpreted as inactive.  Since the values help determine if
        call should be or should remain queued for ACB, the calling
        party's identity should be used within the calculation when
        appropriate.

        The invoke to provide_value for callfwdactive does not contain
        elements cfu, cfbl, cfda, and scf since it is a request to
        obtain such data.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="cfu" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="notsupported"/>
          <xs:enumeration value="active"/>
          <xs:enumeration value="inactive"/>
          <xs:enumeration value="spare"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="cfbl" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="notsupported"/>
          <xs:enumeration value="active"/>
          <xs:enumeration value="inactive"/>
          <xs:enumeration value="spare"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="cfda" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="notsupported"/>
          <xs:enumeration value="active"/>
          <xs:enumeration value="inactive"/>
          <xs:enumeration value="spare"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="scf" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="notsupported"/>
          <xs:enumeration value="active"/>
          <xs:enumeration value="inactive"/>
          <xs:enumeration value="spare"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

```
<xs:simpleType name="termrestrictType">
  <xs:annotation>
    <xs:documentation>
       The termrestrict indicates the termination restriction status
       (or request for status) of called (invoked) party.  The
       following are the potential status values: denied, unrestricted,
       scr, vis, and spare.  Value denied indicates that a user cannot
       receive any terminations because of a service.  Value
       unrestricted indicates that the user is not restricting the call
       or that such restrictions are currently unknown or
       indeterminate.  Value scr (selective call restriction) indicates
       that the call is restricted.  Value vis indicates that it is a
       restricted call; however the meaning is not clearly indicated
       within the mentioned "ACB PROTOCOL, TECHNICAL REQUIREMENTS"
       specification.  Unless spare is known to reflect otherwise, it
       should be interpreted as unrestricted.  Since the values help
       determine if call should be or should remain queued for ACB, the
       calling party's identity should be used within the calculation
       when appropriate.

       The empty string is for use within invoke to request
       termrestrict status.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value=""/>
    <xs:enumeration value="denied"/>
    <xs:enumeration value="unrestricted"/>
    <xs:enumeration value="scr"/>
    <xs:enumeration value="vis"/>
    <xs:enumeration value="spare"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="dntolineType">
  <xs:annotation>
    <xs:documentation>
       The dntoline provides an indication of match and line service
       type.   The match indicates if invoked (called) party was the
       reached (matched) party: match, nomatch, or spare.  If spare
       value is unknown, it should be treated as nomatch.  The line
       service type indicates the type of invoked party.  The values
       are from section 3.4.8 of the mentioned "ACB PROTOCOL, TECHNICAL
       REQUIREMENTS" specification; however the document does not
       define their meaning.

       The match and type are used to help determine if the acb request
       should be or remain queued.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="match" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="match"/>
          <xs:enumeration value="nomatch"/>
         <xs:enumeration value="spare"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="type" minOccurs="0" maxOccurs="1">
      <xs:simpleType>
```

```
          <xs:restriction base="xs:string">
            <xs:enumeration value="individual"/>
            <xs:enumeration value="coin"/>
            <xs:enumeration value="hunt"/>
            <xs:enumeration value="unassigned"/>
            <xs:enumeration value="pbx"/>
            <xs:enumeration value="multiparty"/>
            <xs:enumeration value="choke"/>
            <xs:enumeration value="nonspecific"/>
            <xs:enumeration value="oos"/>
            <xs:enumeration value="telecampus"/>
            <xs:enumeration value="isdn"/>
            <xs:enumeration value="telekibbutz"/>
            <xs:enumeration value="spare"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

3.42.2.4 Example Message

The following is a SUBCRIBE example reflecting the new event package and content-type.

```
SUBSCRIBE sip:+13015551111@pstn.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.40.10;
  branch=z9hG4bKBroadWorks.-1t1hjce-192.168.40.11V5060-0-9-91-124390;
  event=legacy-acb
From:<sip:+13015550000@broadsoft.com;user=phone>;tag=1
To:<sip:+13015551111@pstn.com;user=phone>
Call-ID:BW131909796040500-565063280@as.broadsoft.com
Cseq:929489740 SUBSCRIBE
Contact:<sip:ascluster.broadsoft.com>
Max-Forwards:10
Event:legacy-acb
Expires:10
Content-Type:application/acb+xml
Content-Length:  ---

<?xml version="1.0"?>
<acbtcap>
  <component id="1">
    <invoke opcode="queue_call">
      <parameters/>
    </invoke>
  </component>
  <component id="2">
    <invoke opcode="provide_value">
      <parameters>
        <busyfreestatus/>
        <callfwdactive/>
        <termrestrict/>
        <dntoline/>
      </parameters>
    </invoke>
  </component>
</acbtcap>
```

3.43 Transparent Proxying of SIP Headers and Options

Cisco BroadWorks operates as a Back-to-Back User Agent (B2BUA) and normally does not proxy SIP headers it does not recognize, that is, “unknown” headers. Similarly, it does not proxy unrecognized options tags found in the *Require* and *Supported* headers.

However, it is possible to configure Cisco BroadWorks to transparently proxy some or all unknown SIP headers. It is also possible to configure Cisco BroadWorks to transparently proxy some or all unknown SIP options in *Supported* and *Require* headers.

In contrast to the unknown SIP headers and option tags, a number of SIP headers and option tags are considered “known” to Cisco BroadWorks. Cisco BroadWorks accepts and processes these headers and tags in incoming SIP messages, and it may send them in outgoing SIP messages. However, Cisco BroadWorks always generates these headers and tags anew as a User Agent Client (UAC) rather than transparently proxy them. (In some cases, the value in the incoming message and outgoing message may be the same, yielding the *appearance* of transparently proxying.) Generally, it is not possible to configure Cisco BroadWorks to transparently proxy these known headers and option tags.

Cisco BroadWorks considers the following headers to be known headers, which cannot be transparently proxied.

Accept	*Expires*	*Reason*
Accept-Encoding	*From*	*Record-Route*
Accept-Language	*History-Info*	*Referred-By*
AccessCode	*Max-Forwards*	*Refer-To*
Alert-Info	*MIME-Version*	*Remote-Party-ID*
Allow	*Min-Expires*	*Replaces*
Allow-Events	*Min-SE*	*Require*
Anonymity	*P-Access-Network-Info*	*Retry-After*
Authentication-Info	*P-Asserted-Identity*	*Route RPID-Privacy*
Authorization	*P-Broadsoft-MSSGatewayAddress*	*Rseq*
BBWE-Orig-Via	*P-BroadWorks-Endpoint-Owner-ID*	*Session*
Call-ID	*P-Called-Party-ID*	*Session-Expires*
Call-Info	*P-Charging-Function-Addresses*	*Subject*
CC-Diversion	*P-Charging-Vector*	*Subscription-State*
Charge	*P-Early-Media*	*Supported*
Contact	*P-Preferred-Identity*	*To*
Content-Disposition	*Priority*	*Unsupported*
Content-Encoding	*Privacy*	*Via*
Content-ID	*Proxy-Authenticate*	*Warning*
Content-Language	*Proxy-Authorization*	*WWW-Authenticate*
Content-Length	*Proxy-Require*	*X-BroadWorks-App-ID*
Content-Type	*P-Served-User-Identity*	*X-BroadWorks-DGC*
Cseq	*Rack*	*X-Nortel-Profile*
Diversion		*X-Origin-IP*
Event		

Cisco BroadWorks considers the following headers to be known headers; however, they may also be transparently proxied like unknown headers.

Accept-Contact
Request-Disposition

Cisco BroadWorks considers the following tags to be known tags, which cannot be transparently proxied.

100rel
199
pref
timer
eventlist
early-session
broadworkscalltypequery

NOTE: Cisco BroadWorks does not perform a validation of transparently proxied headers and options. This capability must be used with care as transparent proxying may violate semantics implied by the proxied elements.

In the basic header proxying scenario, Cisco BroadWorks receives a header in an incoming INVITE request and copies it to the outgoing INVITE request, according to its configured header proxying policies. This basic scenario is shown in the call flow diagram in the following figure. The *User-to-User* header is an unknown header, and Cisco BroadWorks is configured to proxy it transparently to Device B. Therefore, Cisco BroadWorks copies the *User-to-User* header from the incoming INVITE request to the outgoing INVITE request.

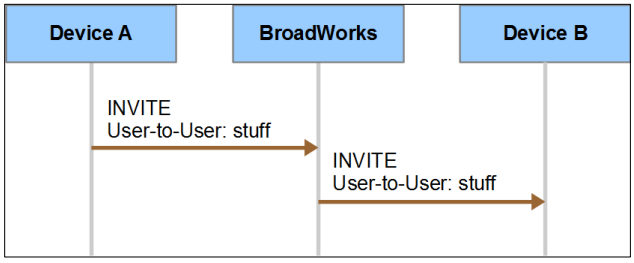

Figure 71 Transparent Proxying of an Unrecognized SIP Header

Cisco BroadWorks' header proxying policies offer some flexibility to control when an unknown header or option tag is proxied. For example, referring again to *Figure 71*, Cisco BroadWorks can be configured to proxy the *User-to-User* header to Device B, depending on whether Device B is considered a device on the access side or the network side, as well as on other criteria. This flexibility is further illustrated in the call flow diagram in *Figure 72*. As seen in this diagram, Cisco BroadWorks proxies the *User-to-User* header to Device B but not to Device C. This behavior is possible, if, for example, the called Cisco BroadWorks user has Simultaneous Ring enabled, Device B is the user's primary access device (for example, an office phone), and Device C is a network device (for example, a home phone).

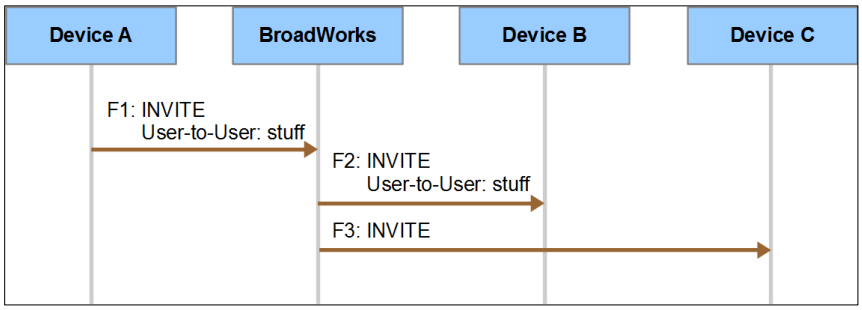

Figure 72 Transparent Proxying Depending on Destination.

Cisco BroadWorks' header proxying policies can be configured to allow SIP devices to "inject" unknown headers when redirecting an INVITE request. There are two basic scenarios in which this header injection is allowed. The first scenario is shown in the call flow diagram in *Figure 73*. Device B sends a *302 Moved Temporarily* response to Cisco BroadWorks to redirect the incoming call. The *Contact* header URI in the *302* response has an embedded *User-to-User* header. Cisco BroadWorks is configured to allow injection of the *User-to-User* header, as well as retention of the header on redirection and egress of the header to Device C, and copies it into the outgoing INVITE request to Device C for the redirection.

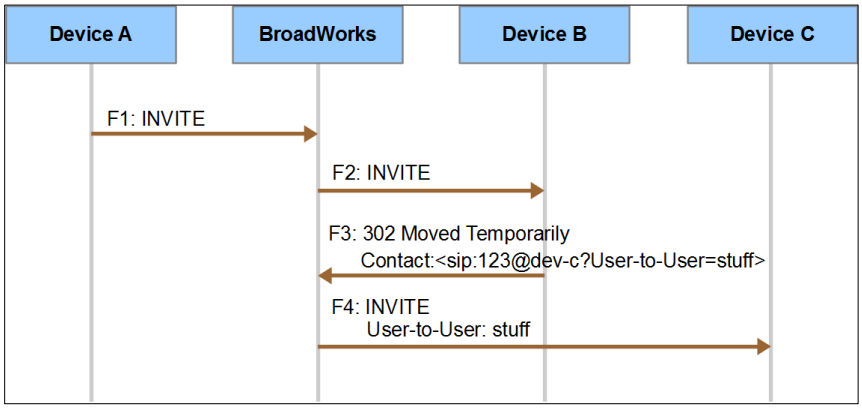

Figure 73 Header Injection from 302 Response.

When Cisco BroadWorks injects an unknown header from a *302* response, as in the scenario just described, it is possible that the incoming INVITE request also had the same header. For example, in *Figure 73*, the INVITE request (F1) from Device A could have a *User-to-User* header. In such a situation, Cisco BroadWorks replaces the header in the incoming INVITE request with the injected header from the *302* response. Although the correct behavior might be instead to merge the two headers, it is impossible for Cisco BroadWorks to know the correct behavior because it does not understand the syntax of the unrecognized header.

The second header injection scenario is shown in the call flow diagram in *Figure 74*, Device A and Device B have an established call. Then Device B sends a REFER request to Cisco BroadWorks to redirect the call to a location at Device C. The *Refer-To* header URI in the REFER request has an embedded *User-to-User* header. Cisco BroadWorks is configured to allow injection of the *User-to-User* header, as well as retention of the header on redirection and egress of the header to Device C, and copies it into the outgoing INVITE request to Device C for the redirection.

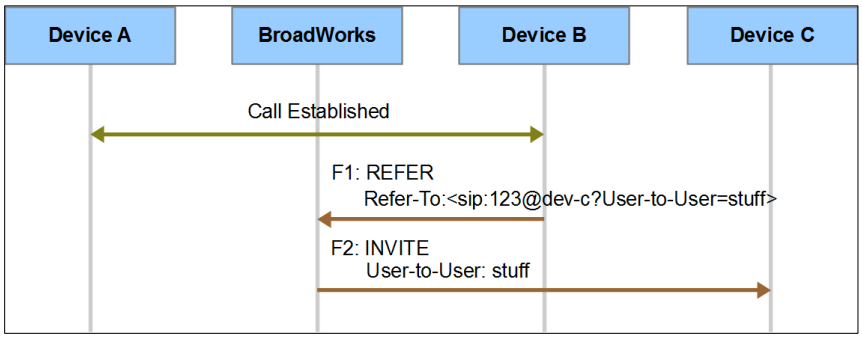

Figure 74 Header Injection from REFER Request.

3.44 Cisco BroadWorks Proprietary Headers

Cisco BroadWorks uses proprietary headers to resolve some complex service interactions over the network interface.

- The *X-BroadWorks-DGC* header is used to carry Cisco BroadWorks proprietary information in some intra-group and intra-enterprise call scenarios.
- The *X-BroadWorks-DNC* header is used to carry Cisco BroadWorks proprietary information over network calls, and is used by the remote network element if it is itself another Cisco BroadWorks Application Server.
- The *P-BroadWorks-Endpoint-Owner-ID* header carries the user ID of the Cisco BroadWorks user who owns the originating endpoint. This information is necessary for services such as Hoteling and Flexible Seating.
- The *X-BroadWorks-SCF* header carries proprietary information between the Service Control Function Server and the Application Server.
- The *X-BroadWorks-Media-ID*, which the Application Server sends to the Media Server or Video Server, contains the media-identifier string to identify all connections in a related call or conference.

It is possible for a message to contain both the *X-BroadWorks-DGC* and *X-BroadWorks-DNC* headers.

These headers should be proxied transparently between two Cisco BroadWorks Application Servers by any intermediate network elements.

NOTE: These headers contain information that has privacy and security considerations. They should not be proxied outside the trust domain, and they should not be proxied to an access device.

3.44.1 Syntax

The following ABNF grammar extends the ABNF in *RFC 3261* and defines the syntax for the *X-BroadWorks-DGC* header.

```
X-BroadWorks-DGC = "X-BroadWorks-DGC" HCOLON [ dgc-param
    *( SEMI dgc-param ) ]

dgc-param = dgc-general-param / dgc-broadworks-anywhere-param /
    dgc-call-center-param / dgc-call-center-calls-param /
    dgc-whisper-param / dgc-call-park-param / dgc-trunk-group-param /
```

```
    dgc-meet-me-conf-param / dgc-executive-param /
    dgc-ea-param / dgc-voice-portal-param / dgc-call-logs-param /
    dgc-mobility-param / dgc-route-list-param / dgc-collaborate-param

; General DGC parameters
dgc-general-param = user-id-param / user-alt-address-param /
    pres-id-param / colp-user-id-param / colp-user-alt-address-param /
    colp-pres-id-param / actual-privacy-param / term-user-id-param /
    alt-address-param / diversion-inhibited-param /
    ext-tracking-id-param / orig-ext-tracking-id-param / ofr-param /
    network-inhibited-param / zone-change-inhibited-param /
    forked-param / answer-after-param / busy-camp-on-param

; BroadWorks Anywhere parameters
dgc-broadworks-anywhere-param = ba-calling-address-param /
     ba-dest-address-param

; Call Center parameters
dgc-call-center-param = cc-wait-time-param / cc-offer-time-param /
    cc-address-param / agent-escalation-param / agent-start-time-param /
    cc-overflow-param / dnis-id-param / colp-dnis-id-param

; Call Center calls parameters
dgc-call-center-calls-param = cccall-cc-user-id-param /
    cccall-cc-name-param / cccall-nb-calls-queue-param /
    cccall-longest-wait-time-param

; Whisper Indicator parameters
dgc-whisper-param = whisper-param / whisper-start-time-param /
    whisper-stop-time-param

; Call Park parameters
dgc-call-park-param = parked-against-user-id-param

; Trunk Group parameters
dgc-trunk-group-param = tg-id-param / colp-tg-id-param /
    tg-pres-id-param / colp-tg-pres-id-param / tg-term-param /
    colp-tg-term-param

; Meet-Me Conferencing parameters
dgc-meet-me-conf-param = meetme-conference-id-param /
    meetme-moderator-param

; Executive Service parameters
dgc-executive-param = exec-param

; Executive-Assistant Service parameters
dgc-ea-param = ea-dest-address-param

; Voice Portal/Voice Mail parameters
dgc-voice-portal-param = vmr-fac-invocation-param

; Enhanced Call Logs parameters
dgc-call-logs-param  = ecl-log-id-param

; BroadWorks Mobility parameters
dgc-mobility-param = mobility-number-param / colp-mobility-number-param

; Route List parameters
dgc-route-list-param = route-list-dn-param / colp-route-list-dn-param

; Collaborate parameters
```

```
dgc-collaborate-param = collaborate-roomid-param /
    collaborate-owner-param

; Individual parameter syntax

user-id-param = "user-id" EQUAL quoted-string

user-alt-address-param = "user-alt-address" EQUAL gen-value

pres-id-param = "pres-id" EQUAL quoted-string

colp-user-id-param = "colp-user-id" EQUAL quoted-string

colp-user-alt-address-param = "colp-user-alt-address" EQUAL gen-value

colp-pres-id-param = "colp-pres-id" EQUAL quoted-string

actual-privacy-param = "actual-privacy" EQUAL "full" / "name" /
    "uri" / "off" / token / quoted-string

term-user-id-param = "term-user-id" EQUAL quoted-string

alt-address-param = "alt-address" EQUAL gen-value

diversion-inhibited-param = "diversion-inhibited"

ext-tracking-id-param = "ext-tracking-id" EQUAL quoted-string

orig-ext-tracking-id-param = "orig-ext-tracking-id" EQUAL quoted-string

ofr-param = "ofr"

network-inhibited-param = "network-inhibited"

zone-change-inhibited-param = "zone-change-inhibited"

forked-param = "forked"

answer-after-param = "answer-after" EQUAL 1*DIGIT

ba-calling-address-param = "ba-calling-address" EQUAL gen-value

ba-dest-address-param = "ba-dest-address" EQUAL gen-value

cc-wait-time-param = "cc-wait-time" EQUAL 1*DIGIT

cc-offer-time-param = "cc-offer-time" EQUAL 1*DIGIT

cc-address-param = "cc-address" EQUAL ["+"] 1*DIGIT

agent-escalation-param = "agent-escalation" EQUAL gen-value

agent-start-time-param = "agent-start-time" EQUAL 1*DIGIT

cc-overflow-param = "cc-overflow" EQUAL cc-overflow-type

cc-overflow-type = "size" / "time"

dnis-id-param = "dnis-id" EQUAL quoted-string

colp-dnis-id-param = "colp-dnis-id" EQUAL quoted-string
```

```
cccall-cc-user-id-param = "cccall-cc-user-id" EQUAL quoted-string

cccall-cc-name-param = "cccall-cc-name" EQUAL quoted-string

cccall-nb-calls-queue-param = "cccall-nb-calls-queue" EQUAL 1*DIGIT

cccall-longest-wait-time-param = "cccall-longest-wait-time" EQUAL 1*DIGIT

whisper-param = "whisper"

whisper-start-time-param = "whisper-start-time" EQUAL 1*DIGIT

whisper-stop-time-param = "whisper-stop-time" EQUAL 1*DIGIT

parked-against-user-id-param = "parked-against-user-id" EQUAL quoted-
string

tg-id-param = "tg-id" EQUAL quoted-string

colp-tg-id-param = "colp-tg-id" EQUAL quoted-string

tg-pres-id-param = "tg-pres-id" EQUAL quoted-string

colp-tg-pres-id-param = "colp-tg-pres-id" EQUAL quoted-string

tg-term-param = "tg-term"

colp-tg-term-param = "colp-tg-term"

meetme-conference-id-param = "meetme-conference-id" EQUAL quoted-string

meetme-moderator-param = "meetme-moderator"

exec-param = "exec" EQUAL exec-value

exec-value = "filtered" / "recall"

ea-dest-address-param = "ea-dest-address" EQUAL quoted-string

vmr-fac-invocation-param = "vmr-fac-invocation"

ecl-log-id-param = "ecl-log-id" EQUAL quoted-string

mobility-number-param = "mobility-number" EQUAL ["+"] 1*DIGIT

colp-mobility-number-param = "colp-mobility-number" EQUAL ["+"] 1*DIGIT

route-list-dn-param = "route-list-dn" EQUAL token

colp-route-list-dn-param = "colp-route-list-dn" EQUAL token

collaborate-roomid-param = "collaborate-roomid" EQUAL quoted-string

collaborate-owner-param = "collaborate-owner"

busy-camp-on-param = "busy-camp-on"
```

The following ABNF grammar extends the ABNF in *RFC 3261* and defines the syntax for the *X-BroadWorks-DNC* header.

```
X-BroadWorks-DNC = "X-BroadWorks-DNC" HCOLON [ dnc-value ]
```

```
dnc-value = ( encrypted-params SEMI "enc" *(SEMI dnc-param) )
            / unencrypted-params

encrypted-params = quoted-string ; when decrypted, matches the rule
                                 ; for unencrypted-params

unencrypted-params = dnc-param *(SEMI dnc-param)

dnc-param = dnc-general-param / dnc-security-class-param /
    dnc-client-session-param

; General DNC parameters
dnc-general-param = actual-identity-param / actual-privacy-param /
    colp-network-address-param / network-address-param /
    redirection-reason-param / user-id-param / moh-param /
    sac-id-param / sac-forking-id-param / transfer-inhibited-param /
    bw-hold-param / soa-param / redir-mobile-param / net-ind-param

; Security Classification parameters
dnc-security-class-param = sc-user-class-param

; Client Session parameters
dnc-client-session-param = client-session-info-param

; Individual parameter syntax

actual-identity-param = "actual-identity" EQUAL quoted-string

actual-privacy-param = "actual-privacy" EQUAL ("full" / "name" /
    "uri" / "off" / token / quoted-string)

colp-network-address-param = "colp-network-address" EQUAL quoted-string

network-address-param = "network-address" EQUAL quoted-string

redirection-reason-param = "redirection-reason" EQUAL
    ("attended-transfer" / "automatic-hold-retrieve" / "barge-in" /
     "call-center" / "call-park-retrieve" / "call-pickup" /
     "deflection" / "follow-me" / "hunt-group" / "no-answer" /
     "recall" / "time-of-day" / "unavailable" / "unconditional" /
     "unknown" / "user-busy" / token)

user-id-param = "user-id" EQUAL quoted-string

moh-param = "moh"

sac-id-param = "sac-id" EQUAL gen-value

sac-forking-id-param = "sac-forking-id" EQUAL gen-value

transfer-inhibited-param = "transfer-inhibited"

bw-hold-param = "bw-hold"

soa-param = "soa"

redir-mobile-param = "redir-mobile"

sc-user-class-param = "sc-user-class" EQUAL quoted-string

client-session-info-param = "client-session-info" EQUAL quoted-string
```

```
net-ind-param = "net-ind" EQUAL net-ind-value

net-ind-value = ("InterNetwork" / "InterHostingNE" / "InterAS" /
"IntraAS")
```

The following ABNF grammar extends the ABNF in *RFC 3261* and defines the syntax for the *P-BroadWorks-Endpoint-Owner-ID* header.

```
P-BroadWorks-Endpoint-Owner-ID = "P-BroadWorks-Endpoint-Owner-ID" HCOLON
    gen-value
```

The following ABNF grammar extends the ABNF in *RFC 3261* and defines the syntax for the *X-BroadWorks-SCF* header.

```
X-BroadWorks-SCF = "X-BroadWorks-SCF" HCOLON
    scf-param *(SEMI scf-param)
    ; list of parameters separated by semicolons

scf-param = imsi-param / call-ref-param / vlr-param /
    msrn-lookup-error-param / msrn-lookup-error-info-param /
    forwarding-reason-param  / generic-param

imsi-param = "imsi" EQUAL gen-value

call-ref-param = "call-ref" EQUAL gen-value

vlr-param = "vlr" EQUAL gen-value

msrn-lookup-error-param = "msrn-lookup-error" EQUAL gen-value

msrn-lookup-error-info-param = "msrn-lookup-error-info" EQUAL gen-value

forwarding-reason-param = "forwarding-reason" EQUAL gen-value
```

The following ABNF grammar extends the ABNF in *RFC 3261* and defines the syntax for the *X-BroadWorks-Media-ID* header.

```
X-BroadWorks-Media-ID = "X-BroadWorks-Media-ID" HCOLON media-identifier

media-identifier = token
```

3.45 Advice of Charge

Cisco BroadWorks supports Advice of Charge (AoC) information on the network interface, specifically AOC-D (during a call). Cisco BroadWorks uses this information to provide Advice of Charge to access devices. AoC must be encoded in the message body of a SIP INFO message. The body must be of type *application/vnd.etsi.aoc+xml,* as defined in *3GPP TS 24.647 v8.0.0 Advice of Charge (AoC)* [51]. Following is an example message.

```
Content-Type:application/vnd.etsi.aoc+xml
Content-Length:362

<?xml version="1.0" encoding="UTF-8"?>
<aoc>
```

```
  <aoc-d>
    <charging-info>subtotal</charging-info>
    <recorded-charges>
      <recorded-currency-units>
        <currency-id>EUR</currency-id>
        <currency-amount>10</currency-amount>
      </recorded-currency-units>
    </recorded-charges>
    <billing-id>normal-charging</billing-id>
  </aoc-d>
</aoc>
```

Cisco BroadWorks also supports tariff information received in SIP *1XX* responses, SIP INFO, or *200 OK* responses to INVITE messages from upstream network equipment. This tariff information must be encoded in a message body of type *application/vnd.etsi.sci+xml*, as defined in *3GPP TS 29.658 v2.1.0 SIP Transfer of IP Multimedia Service Tariff Information* [52]. Following is an example message.

```
Content-Type:application/vnd.etsi.sci+xml
Content-Length: 2400

<?xml version="1.0" encoding="UTF-8"?>
<messageType xmlns="http://uri.etsi.org/ngn/params/xml/simservs/sci">
  <crgt>
    <chargingControlIndicators>
      <immediateChangeOfActuallyAppliedTariff>true
      </immediateChangeOfActuallyAppliedTariff>
    </chargingControlIndicators>
    <chargingTariff>
      <tariffPulse>
        <currentTariffPulse>
          <communicationChargeSequencePulse>
            <pulseUnits>01</pulseUnits>
            <chargeUnitTimeInterval>1100</chargeUnitTimeInterval>
            <tariffDuration>10</tariffDuration>
          </communicationChargeSequencePulse>
          <tariffControlIndicators>true</tariffControlIndicators>
          <callAttemptChargePulse>03</callAttemptChargePulse>
          <callSetupChargePulse>0A</callSetupChargePulse>
        </currentTariffPulse>
      </tariffPulse>
    </chargingTariff>
    <originationIdentification>
      <networkIdentification>022222</networkIdentification>
      <referenceID>36</referenceID>
    </originationIdentification>
    <destinationIdentification>
      <networkIdentification>023333</networkIdentification>
      <referenceID>57</referenceID>
    </destinationIdentification>
    <currency>USD</currency>
  </crgt>
</messageType>
```

NOTE: Cisco BroadWorks accepts *application/vnd.etsi.aoc+xml* and *application/vnd.etsi.sci+xml* body types from the network and uses them to provide AoC information to access devices when the service is enabled. However, Cisco BroadWorks never sends these message bodies on the network interface.

3.46 P-Camel Headers

The Cisco BroadWorks Application Server retrieves information from the following SIP headers for accounting purposes:

- *P-CAMEL-Loc-Info*
- *P-CAMEL-MSC-Address*
- *P-CAMEL-CellIdOrLAI*

Cisco BroadWorks does not proxy these headers.

3.47 Calling Line ID Unavailable and Anonymous

When receiving INVITEs from the network for termination to Cisco BroadWorks users, Cisco BroadWorks uses the following indications to determine whether the calling party identity is unavailable or private. First, the identity is determined by selecting one of the following headers in decreasing order of priority:

- *P-Preferred-Identity*
- *P-Asserted-Identity*
- *Remote-Party-ID*
- *From*

Then the message is checked for an unavailable or private signature. *Table 3* shows the different signaling signatures recognized as unavailable calling identities.

Unavailable Signature	Example
User part of identity SIP URI is “unknown”.	From: sip:**unknown**@host;tag=123
No user part in identity.	From: sip:**host**;tag=123
Identity taken from the *From* header with display name set to “unknown”	From: **"unknown"**<sip:user@host>;tag=123
Identity taken from Remote-Party-ID with “screen” **not** set to “yes” (or absent). **NOTE**: If *privacyEnforceScreening* is “false”, then this signature does not apply and is ignored.	Remote-Party-ID: "name"<sip:user@host>;**screen=no**
Received calling party category value for operator or pay phone. **NOTE**: If *extendedCallingLineID* is “false”, then this signature does not apply and is ignored.	From: "name"<sip:+15145550100;**cpc=payphone**@host;user=phone>;tag=123

Table 3 Unavailable Identity Signatures

Similarly, *Table 4* shows the signaling signatures recognized as unavailable calling identities.

Private Signature	Example
Privacy header with “user” or “id” setting.	Privacy: **user**
Identity taken from Remote-Party-ID with “screen” set to “yes” and *Anonymity* header with “uri” setting. **NOTE**: If *privacyEnforceScreening* is “false”, then the “screen” condition is ignored and the signature applies even if “screen” is missing or set to a different value.	Remote-Party-ID: "name"<sip:user@host>;**screen=yes** Anonymity: **uri**

Private Signature	Example
Identity taken from Remote-Party-ID with screen set to “yes” and *RPID-Privacy* header with “privacy” set to “full” or “uri”. **NOTE**: If *privacyEnforceScreening* is “false”, then the “screen” condition is ignored and the signature applies even if “screen” is missing or set to a different value.	Remote-Party-ID: "name"<sip:user@host>;**screen=yes** RPID-Privacy: **privacy=full**
Identity taken from Remote-Party-ID with screen set to “yes” and “privacy” set to “full” or “uri”. **NOTE**: If *privacyEnforceScreening* is “false”, then the “screen” condition is ignored and the signature applies even if “screen” is missing or set to a different value.	Remote-Party-ID: "name"<sip:user@host>; **screen=yes**;**privacy=uri**
Identity taken from the *From* header with user part of identity SIP URI set to “restricted”.	From: sip:**restricted**@host;tag=123
Identity taken from the *From* header with user part of identity SIP URI set to “anonymous”.	From: sip:**anonymous**@host;tag=123
The user part of the selected identity SIP URI does not represent a phone number AND no privacy has been found via any other mean described in this document.	From: sip:**urldialing**@host;tag=123

Table 4 Private Identity Signatures

3.48 Call Recording

Cisco BroadWorks provides extensions to support Call Recording. These extensions apply only between Cisco BroadWorks and the Call Recording Server.

For details about the call recording solution, see the *Cisco BroadWorks Call Recording Interface Guide* [67].

3.49 Priority, Resource-Priority, and P-DCS-OSPS SIP Headers for Emergency Calls

In North America, emergency calling has some distinct functionality that allows an operator to identify and communicate with the calling party who is making an emergency call. Cisco BroadWorks supports emulation of the following circuit-switched emergency calling services:

- *Network or Bureau Hold* – Enables the operator to maintain control of the call, and notifies the operator upon calling party disconnect.
- *Operator Ring-back* – Allows the operator to re-establish communication with the calling party in cases when the calling party has gone on-hook, or remains off-hook, but has become unresponsive.
- *Forced Disconnect* – Allows the operator to disconnect the call.

To support this feature, the emergency originating endpoint device sends either of the following two SIP headers:

- *Resource-Priority: emgr.0* (defined in *RFC 4412* [61])
- *Priority: emergency* (defined in *RFC 3261* [1])

Cisco BroadWorks proxies the received header if the called number has been identified as an emergency number.

This is an example of an initial INVITE with *Resource-Priority* and *Priority* header.

```
INVITE sip:911@10.16.129.6 SIP/2.0
Via: SIP/2.0/UDP 10.16.129.4;branch=z9hG4bKBroadWorks.1ssl2nc-
Contact: <sip:10.16.129.4>
To: <sip:911@txasdev80.rtx.broadsoft.com>
From:<sip:9726980601@10.16.129.4;user=phone>;tag=394435134-1308772446
Call-ID:BW145406658220611-39230356 4@10.16.129.4
CSeq: 1 INVITE
Resource-Priority: emgr.0
Priority: emergency
Max-Forwards: 5
.
.
```

During an emergency call, it is desirable for the originating device not to release the call when going on hook. So upon hang up (handset on-hook), the SIP endpoint devices with *Supports Emergency Disconnect Control* enabled can be configured to not send a BYE to terminate the call. Instead, these SIP endpoint devices send a re-INVITE with a=inactive in the SDP when the handset is put on-hook. This is interpreted by the Application Server as a call-on-hold request.

A disconnect tone is played from the Media Server toward the emergency operator when the emergency call is put on hold (hang up) by the originator. This tone is played until a re-INVITE with the *P-DCS-OSPS: RING* header is received from the terminating emergency operator. The disconnect tone is stopped and a re-INVITE is sent to the emergency originator. The originating phone should be ringing upon receiving this re-INVITE.

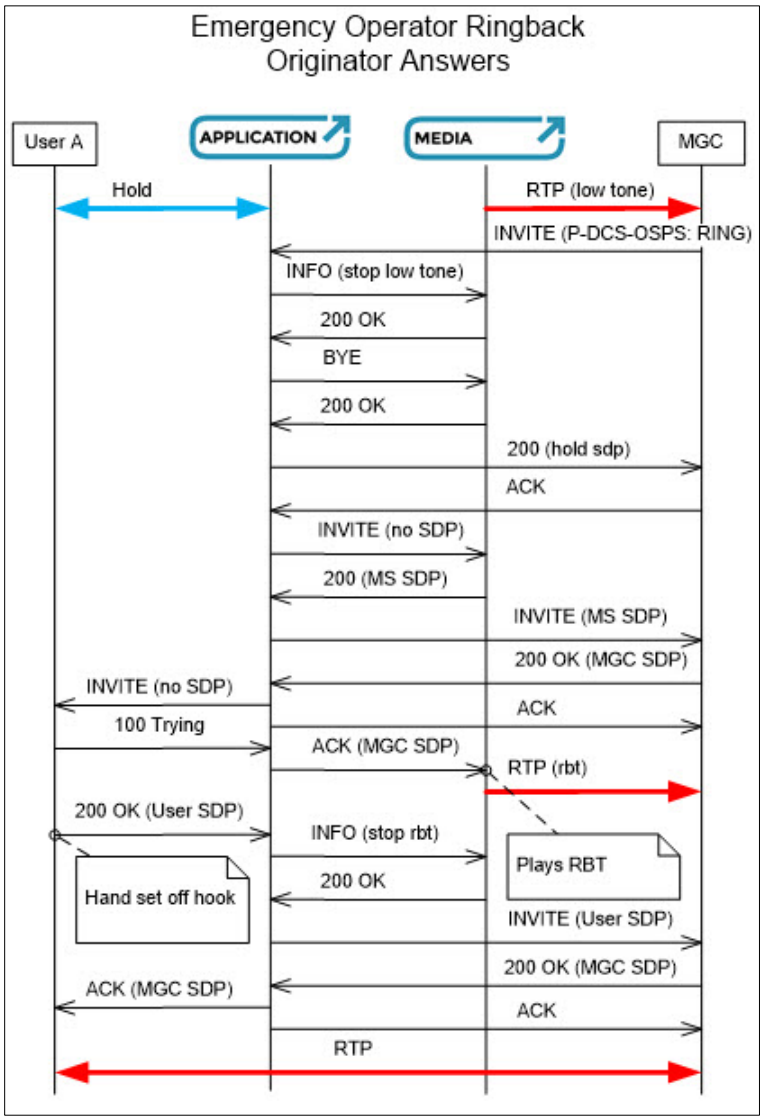

Figure 75 Emergency Operator Rings Originating Phone

The network-side interface now accepts the *P-DCS-OSPS* header with value set to "RING" in a re-INVITE. Any other value in the *P-DCS-OSPS* is ignored. The *P-DCS-OSPS* header is defined in *RFC 5503* [60].

This is an example of a re-INVITE with *P-DCS-OSPS* header.

```
INVITE sip:911@txasdev80.rtx.broadsoft.com SIP/2.0
Via:SIP/2.0/UDP 10.16.129.6;branch=z9hG4bKBroadWorks.1ssl2nc-
Contact: <sip:10.16.129.6>
To: <sip:911@txasdev80.rtx.broadsoft.com>;tag=394435134-1adfadfee3
From:<sip:9726980601@10.16.129.4;user=phone>;tag=394435134-1308772446
Call-ID:BW145406658220611-39230356 4@10.16.129.4
CSeq: 1 INVITE
P-DCS-OSPS: RING
Max-Forwards: 5
.
.
.
```

3.50 IPv6 and IPv4/IPv6 Dual Stack Support (RFC 6947)

Cisco BroadWorks' SIP interface operates in one of three different modes: IPv4 only, IPv6 only, or dual-stack mode. In dual-stack mode, Cisco BroadWorks supports IPv4 and IPv6 simultaneously.

In all three modes, Cisco BroadWorks can successfully parse the IPv6 address syntax in SIP headers and SDP message bodies. Cisco BroadWorks complies with the IPv6 recommendations found in *RFC 4566* [63], *RFC 5954* [65], and *RFC 5952* [64]. Note that IPv6 normalization to canonical form applies only to addresses generated by Cisco BroadWorks and not to values received and proxied across. Received unnormalized IPv6 addresses proxied across Cisco BroadWorks are not normalized.

Cisco BroadWorks does not support IPv6 scoped addresses, link-local addresses, IPv4-mapped IPv6 addresses, and IPv4-Embedded IPv6 addresses.

The operational mode determines the IP version Cisco BroadWorks uses for its SIP interface. In IPv4-only mode, Cisco BroadWorks accepts SIP messages and sends SIP messages using only IPv4. In IPv6-only mode, Cisco BroadWorks accepts SIP messages and sends SIP messages using only IPv6. In addition, in dual-stack mode, Cisco BroadWorks accepts SIP messages and sends SIP messages using either IPv4 or IPv6.

When sending a SIP request to network elements in dual-stack mode, Cisco BroadWorks must decide whether to use IPv4 or IPv6. To make this decision, Cisco BroadWorks depends on information provisioned in the Network Server. Each routing network element (NE) or hosting NE provisioned in the Network Server has properties that indicate whether it supports IPv4, IPv6, or dual stack. If the NE supports only IPv4 or only IPv6, then Cisco BroadWorks sends the SIP request using IPv4 or IPv6, respectively. If the NE supports both IPv4 and IPv6 (dual stack), then Cisco BroadWorks sends the SIP request using either IPv4 or IPv6.

Cisco BroadWorks supports the alternate connectivity mechanism for media negotiation, as described in *RFC 6947* [66]. In this mechanism, the offer SDP includes one or more "a=altc" lines which propose alternate connection information. Cisco BroadWorks passes these lines transparently between the two remote endpoints, allowing the remote endpoints to successfully negotiate the use of IPv4 or IPv6. When Cisco BroadWorks is operating in dual-stack mode and generates a "hold" SDP as an offer SDP, it adds an "a=altc" line for IPv6 and a second "a=altc" line for IPv4. (The first "a=altc" line indicates the preferred address, so Cisco BroadWorks always prefers IPv6.) Similarly, when the Cisco BroadWorks Media Server is operating in dual-stack mode, it generates these same "a=altc" lines. When Cisco BroadWorks is operating in dual-stack mode and generates a "hold" answer SDP, it again follows the alternate connectivity protocol. More specifically, if the offer SDP has an "a=altc" line, Cisco BroadWorks honors the IP address version in that line as the preference of the remote endpoint, and generates the hold SDP accordingly. For example, if the first "a=altc" line contains an IPv6 address, Cisco BroadWorks generates a hold answer SDP with IPv6 addresses.

NOTE: Cisco BroadWorks follows the precaution regarding alternate connectivity attributes as recommended in *RFC 6947*. A potential problem arises if a middlebox, such as an SBC, changes addresses in the SDP without understanding the alternate connectivity attribute. To identify the problem, Cisco BroadWorks checks that the address in the c= line is also one of the addresses in an "a=altc" line. If this check fails, then Cisco BroadWorks assumes that the original c= line was modified, and Cisco BroadWorks ignores the "a=altc" lines.

Depending on its configuration, Cisco BroadWorks may add the “altc” option to the Supported header.

3.50.1 Message Examples

The following is an example of an INVITE request between two IPv6 nodes.

```
INVITE sip:619@mtlasdev86.net;user=phone SIP/2.0
Via: SIP/2.0/UDP
[fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29]:5050;branch=z9hG4bKc42bcaae0d5ba68
cd8e170425087270b6ec73337,SIP/2.0/UDP
[fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29]:5070;branch=z9hG4bK6201;received=f
d5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29
From: <sip:5146986601@mtlasdev86.net>;tag=6047
To: <sip:619@mtlasdev86.net>
Call-ID: 4489
CSeq: 20 INVITE
Contact: <sip:south01@[fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29]:5070>
Content-Type: application/sdp
Allow: INVITE,ACK,CANCEL,OPTIONS,BYE,REFER,NOTIFY,MESSAGE,SUBSCRIBE,INFO
Max-Forwards: 69
Subject: Phone call
Record-Route: <sip:[fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29]:5050;lr>
P-Asserted-Identity: <sip:5146986601@mtlasdev86.net>
P-Access-Network-Info: IEEE-602.11b; isp=abc601;network=4601
P-Charging-Vector: icid-value=3C4F9488C5469BB9;orig-ioi=radio601.test
P-Charging-Function-Addresses: ccf=rf.mpiotte.mtl.broadsoft.com
Route: <sip:[fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29]:5060;lr;call=orig>,
<sip:[fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29]:5050;lr;call=orig>
Content-Length: 458

v=0
o=5146986601 123456 654321 IN IP6 fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29
s=A conversation
c=IN IP6 fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29
t=0 0
m=audio 7078 RTP/AVP 112 111 110 3 0 8 101
a=rtpmap:112 speex/32000/1
a=fmtp:112 vbr=on
a=rtpmap:111 speex/16000/1
a=fmtp:111 vbr=on
a=rtpmap:110 speex/8000/1
a=fmtp:110 vbr=on
a=rtpmap:3 GSM/8000/1
a=rtpmap:0 PCMU/8000/1
a=rtpmap:8 PCMA/8000/1
a=rtpmap:101 telephone-event/8000/1
a=fmtp:101 0-11
```

The following is an example of an INVITE request with an SDP that supports alternate connectivity.

```
INVITE sip:5146999600@192.168.8.79:5060 SIP/2.0
Via:SIP/2.0/UDP
mtlasdev99.mtl.broadsoft.com;branch=z9hG4bKBroadWorks.1jmomag-
192.168.8.79V5060-0-841556780-1615607705-1330975491719
From:<sip:601@mtlasdev99.net>;tag=1615607705-1330975491719
To:<sip:5146999600@mtlasdev99.net>;tag=2932b69143ed7f5do0
Call-ID:f4285da1-9855838d@192.168.8.79
CSeq:841556780 INVITE
Contact:<sip:mtlasdev99.mtl.broadsoft.com:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
```

```
Supported:altc
Accept:application/media_control+xml,application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:261

v=0
o=BroadWorks 23 0 IN IP4 192.168.8.94
s=-
c=IN IP4 0.0.0.0
t=0 0
m=audio 16476 RTP/AVP 0 101
a=altc IP6 2001::8:b09b:e7ad:2c22:d96f 16476
a=altc IP4 0.0.0.0 16476
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:30
```

3.51 Call Correlation Identifier

To correlate log file entries on Cisco BroadWorks servers with log file entries on other SIP devices, including other Cisco BroadWorks servers, Cisco BroadWorks may be configured to add a unique correlation identifier to certain SIP messages. Cisco BroadWorks populates the correlation identifier into a proprietary *X-BroadWorks-Correlation-Info* header.

The *X-BroadWorks-Correlation-Info* header has the following syntax.

```
X-BroadWorks-Correlation-Info = "X-BroadWorks-Correlation-Info" HCOLON
BW-Correlation-value

BW-Correlation-value = TEXT-UTF8-TRIM
```

Cisco BroadWorks may add the *X-BroadWorks-Correlation-Info* header for SIP messages that meet the following criteria:

- INVITE request outside a dialog (initial INVITE request)
- INVITE request inside a dialog (re-INVITE request)
- 18x response to an INVITE request
- 200 response to an INVITE request
- SUBSCRIBE request for the *calling-name* event package

The Cisco BroadWorks Application Server includes the *X-BroadWorks-Correlation-Info* header in INVITE requests to the Cisco BroadWorks Media Server and Cisco BroadWorks Network Server.

Based on Cisco BroadWorks' configuration, the correlation identifier can be unique within a Cisco BroadWorks cluster or globally unique.

3.52 Stateless Proxy for Geographical Redundancy

3.52.1 Overview

For enhanced reliability, the Cisco BroadWorks platform is deployed as a pair of Application Servers, with one Application Server designated the primary Application Server and the other the secondary Application Server. Both Application Servers support identical functionality. Under normal operating conditions, the primary Application Server processes all calls and other SIP messaging. However, under certain conditions, such as when the primary Application Server is offline for maintenance activity or because of a failure condition, the secondary Application Server may take over and process calls instead of the primary Application Server. This redundancy functionality is described in the *Cisco BroadWorks Redundancy Guide* [42].

The primary Application Server and secondary Application Server each maintain an awareness of the state of the other via a dedicated communication link. Therefore, the secondary Application Server generally knows when the primary Application Server is in a state suitable for processing an incoming call, and vice versa. This awareness enables intelligent processing and routing in the primary and secondary Application Servers for enhanced performance when there is a loss of connectivity. For instance, if the secondary Application Server receives an initial INVITE request from an access device or network device, and if it knows that the primary Application Server is available to process a new call, then it may take the role of a stateless proxy server and route the INVITE request to the primary Application Server. This scenario is depicted in the following figure.

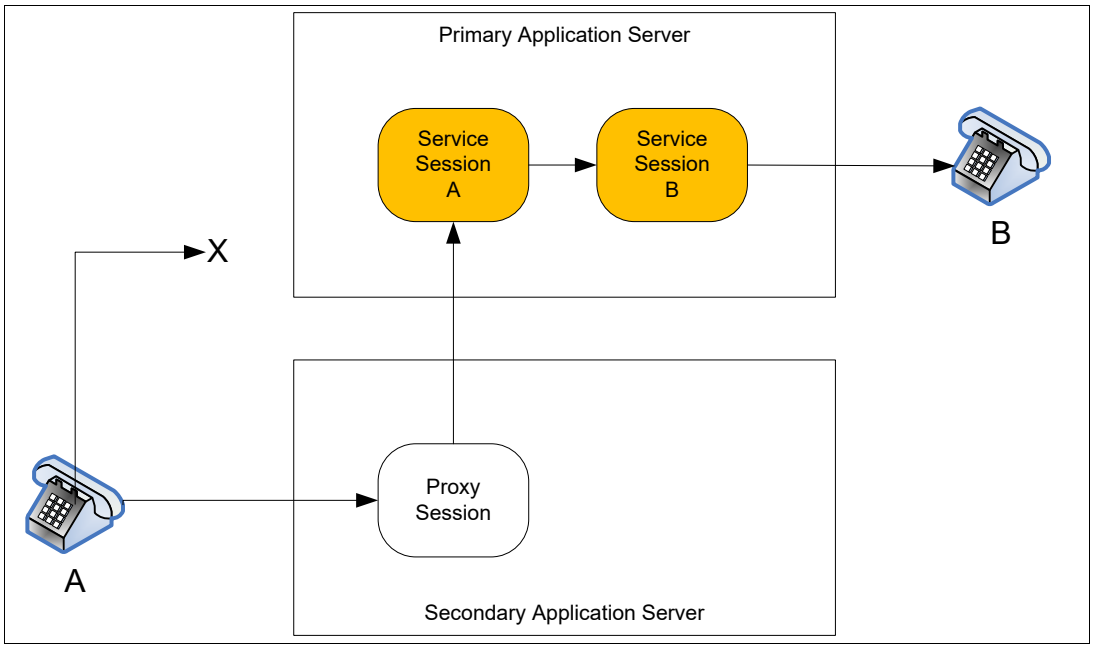

Figure 76 Stateless Proxy Server Scenario

In the previous figure, the phone labeled “A” is an access device which first attempts unsuccessfully to send an INVITE request to the primary Application Server. The access device eventually fails over to the secondary Application Server, which makes a decision to route the INVITE request to the primary Application Server. In this way, the new call proceeds without a loss of functionality, despite a loss of network connectivity between the access device and the primary Application Server.

The proxy server behavior on the secondary Application Server is a configuration option that can be enabled or disabled. If the behavior is disabled, the secondary Application Server always processes an initial INVITE request itself.

3.52.2 Call Flows

There are three different call scenarios in which the secondary Application Server performs as a stateless proxy server.

3.52.2.1 Scenario 1: Access Device to Secondary Application Server to Primary Application Server

The first scenario is shown in the call flow diagram in *Figure* 77. The access device attempts unsuccessfully to send an initial INVITE request (F0) to the primary Application Server. After a short time, the access device sends the INVITE request (F1) to the secondary Application Server. The secondary Application Server, knowing that it has connectivity to the primary Application Server, and that the primary Application Server is able to process calls, routes the INVITE request (F2) as a stateless proxy server to the primary Application Server. The primary Application Server, on receiving the INVITE request, creates a call processing session and handles the call. Note that the SIP signaling between the primary Application Server and the network device is shown for completeness, but is otherwise irrelevant to the interaction of the primary Application Server, secondary Application Server, and access device in this scenario.

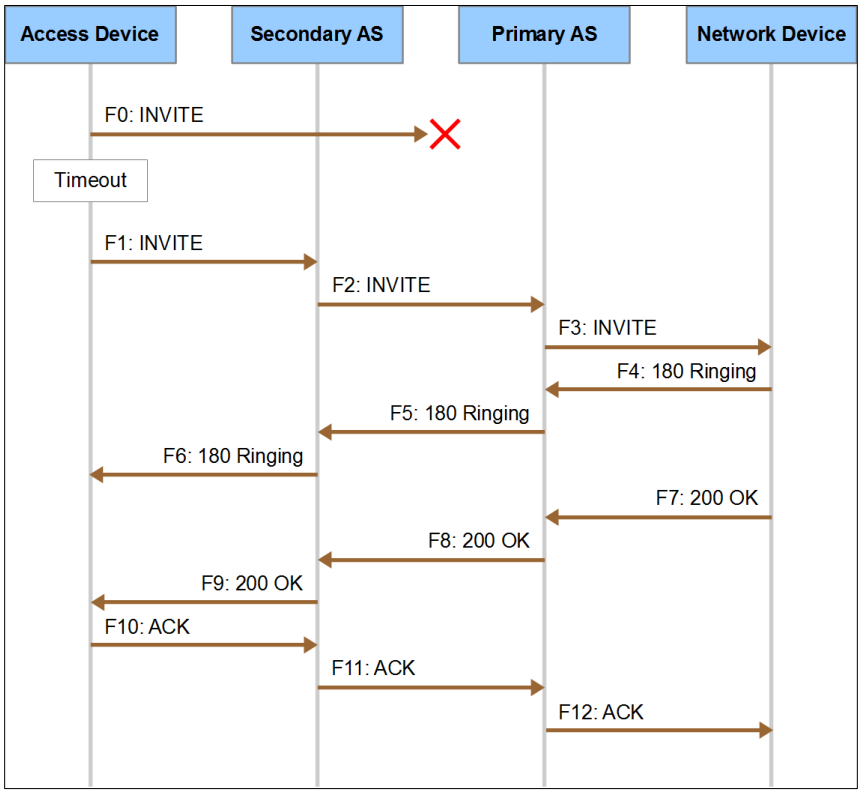

Figure 77 Proxy Scenario: Access Device Cannot Reach Primary Application Server

The secondary Application Server uses the *Record-Route* mechanism to remain in the signaling path for the duration of the SIP dialog. Therefore, the secondary Application Server adds a *Record-Route* header to the INVITE request (F2) to the primary Application Server. The secondary Application Server adds special parameters to its *Record-Route* entry in order to more closely coordinate actions with the primary Application Server. Specifically, the secondary Application Server adds the following parameters:

- *bwgeoproxy* – This parameter tags the entry as one that identifies the secondary Application Server acting in the proxy server role.
- *bwpeer* – This parameter also tags the entry as one that identifies the secondary Application Server acting in the proxy server role. However, it appears only in the leg between the primary and secondary Application Servers.

Following is an example of the *Record-Route* header in the INVITE request (F2).

```
Record-Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer>
```

Moreover, the secondary Application Server adds an *X-BroadWorks-Source* header to the INVITE request (F2). This header contains the IP address from which the secondary Application Server received the initial INVITE request (F1).

Following is an example of the *X-BroadWorks-Source* header in the INVITE request (F2).

```
X-BroadWorks-Source:10.0.40.40
```

Following the procedures in *RFC 3261*, the primary Application Server copies the received *Record-Route* header into the responses (F5, F8) to the INVITE request. The secondary Application Server, however, rewrites its *Record-Route* header entry before forwarding the responses (F6, F9) to the access device. Specifically, the secondary Application Server removes the *bwpeer* parameter, and may rewrite the IP address in the URI.

For this scenario to succeed, the access device must support the *Record-Route* mechanism, so that the secondary Application Server can remain in the signaling path.

3.52.2.2 Scenario 2: Network Device to Secondary Application Server to Primary Application Server

The second scenario, shown in the call flow diagram in *Figure 78*, resembles the first scenario, except that a network device sends the initial INVITE request. The network device attempts unsuccessfully to send an initial INVITE request (F0) to the primary Application Server. After a short time, the network device sends the INVITE request (F1) to the secondary Application Server. The secondary Application Server, knowing that it has connectivity to the primary Application Server, and that the primary Application Server is able to process calls, routes the INVITE request (F2) as a stateless proxy server to the primary Application Server. The primary Application Server, on receiving the INVITE request, creates a call processing session and handles the call. Note that the SIP signaling between the primary Application Server and the access device is shown for completeness, but is otherwise irrelevant to the interaction of the primary Application Server, secondary Application Server, and network device in this scenario.

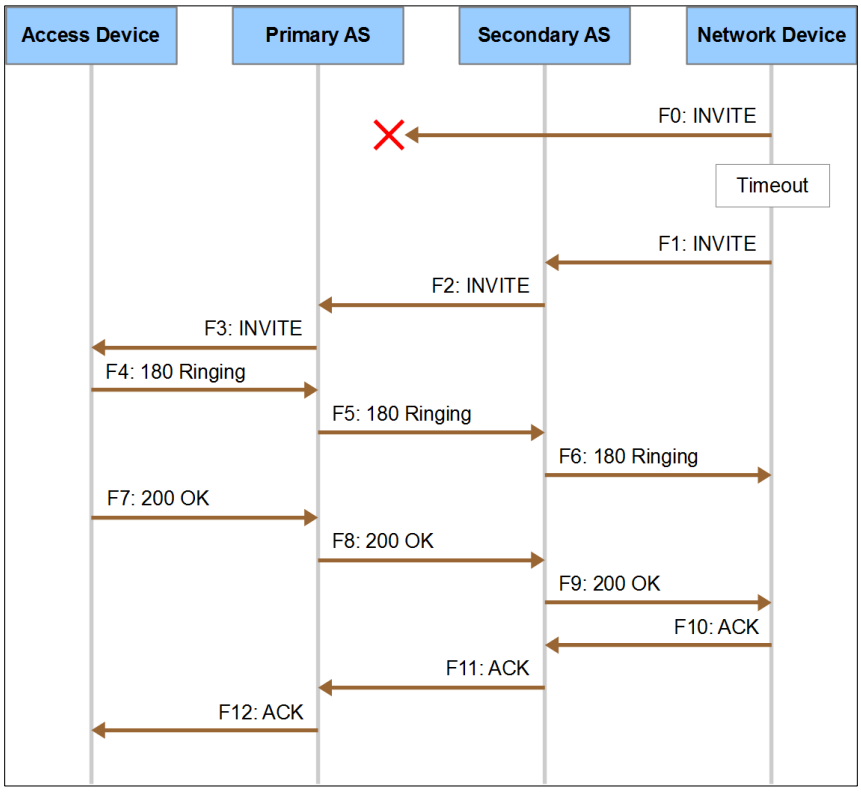

Figure 78 Proxy Scenario: Network Device Cannot Reach Primary Application Server

The secondary Application Server uses the *Record-Route* mechanism to remain in the signaling path for the duration of the SIP dialog. Therefore, the secondary Application Server adds a *Record-Route* header to the INVITE request (F2) to the primary Application Server. The secondary Application Server also adds special parameters to its *Record-Route* entry in order to more closely coordinate actions with the primary Application Server. Specifically, the secondary Application Server adds the following parameters:

- *bwgeoproxy* – This parameter tags the entry as one that identifies the secondary Application Server acting in the proxy server role.
- *bwpeer* – This parameter also tags the entry as one that identifies the secondary Application Server acting in the proxy server role. However, it appears only on the leg between the primary and secondary Application Servers.
- *bwnetwork* – This parameter indicates that the secondary Application Server is interacting with a device on the network side.

The following is an example of the *Record-Route* header in the INVITE request (F2).

```
Record-Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer;bwnetwork>
```

Moreover, the secondary Application Server adds an *X-BroadWorks-Source* header to the INVITE request (F2). This header contains the IP address from which the secondary Application Server received the initial INVITE request (F1).

The following is an example of the *X-BroadWorks-Source* header in the INVITE request (F2).

```
X-BroadWorks-Source:10.0.41.5
```

Following the procedures in *RFC 3261*, the primary Application Server copies the received *Record-Route* header into the responses (F5, F8) to the INVITE request. The secondary Application Server, however, rewrites its *Record-Route* header entry before forwarding the responses (F6, F9) to the access device. Specifically, the secondary Application Server removes the *bwpeer* parameter and may rewrite the IP address in the URI.

For this scenario to succeed, the network device must support the *Record-Route* mechanism, so that the secondary Application Server can remain in the signaling path.

3.52.2.3 Scenario 3: Primary Application Server to Secondary Application Server to Access Device

The third scenario is shown in *Figure 79* . The primary Application Server has just received an initial INVITE request (F1) from the network device. It has completed its call processing and has prepared an initial INVITE request to send to the access device. The primary Application Server believes, based on an internal indicator, that the device endpoint at this access device is unreachable. The primary Application Server also knows that the secondary Application Server is reachable and that it is able to perform the role of a proxy server. Therefore, the primary Application Server sends the INVITE request (F2) to the secondary Application Server, which performs the role of a proxy server and routes the INVITE request (F3) to the access device. Note that the SIP signaling between the primary Application Server and the network device is shown for completeness, but is otherwise irrelevant to the interaction of the primary Application Server, secondary Application Server, and access device in this scenario.

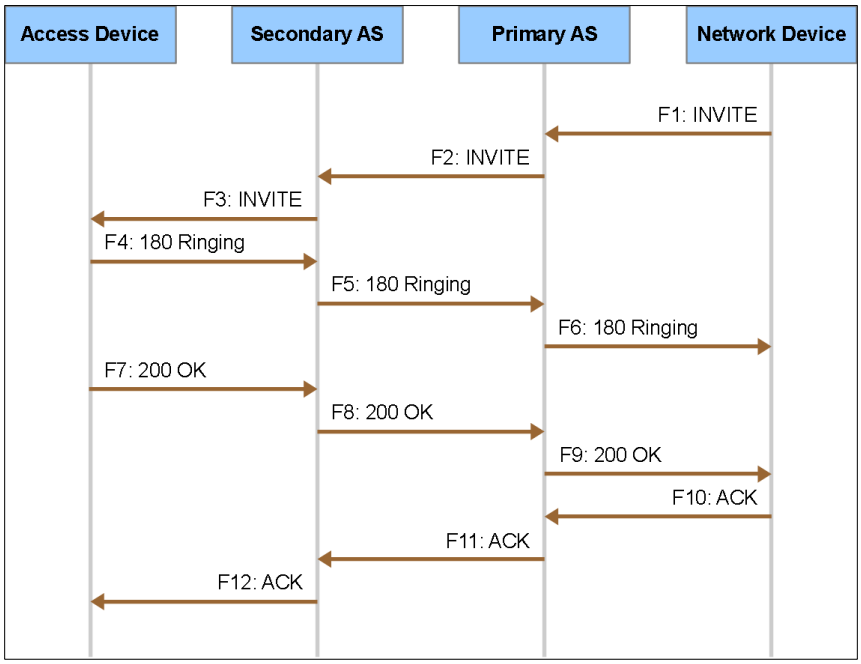

Figure 79 Proxy Scenario: Primary Application Server Cannot Reach Access Device.

The primary Application Server, while performing its call processing, selects the route to the access device and conveys this information to the secondary Application Server in a *Route* header. The INVITE request (F2) from the primary Application Server to the secondary Application Server, therefore, has a *Route* header with two entries. The first entry identifies the secondary Application Server, and it contains special parameters to coordinate actions between the primary and secondary Application Server. These parameters include:

- *bwgeoproxy* – This parameter tags the entry as one that identifies the secondary Application Server acting in the proxy server role.
- *bwpeer* – This parameter also tags the entry as one that identifies the secondary Application Server acting in the proxy server role. However, it appears only the leg between the primary and secondary Application Servers.

The second entry indicates the route to the access device. The secondary Application Server removes this entry before forwarding the INVITE request to the access device. To unambiguously tag this entry, the primary Application Server adds a special *bwdeleteme* parameter.

The following is an example of the *Route* header.

```
Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer>,
  <sip:10.0.40.40:5060;lr;bwdeleteme>
```

The secondary Application Server uses the *Record-Route* mechanism to remain in the signaling path for the duration of the SIP dialog. Therefore, the secondary Application Server adds a *Record-Route* header to the INVITE request (F2) to the access device. The *Record-Route* entry contains a *bwgeoproxy* parameter.

The following is an example of the *Record-Route* header in the INVITE request (F3) from the secondary Application Server to the access device:

```
Record-Route:<sip:192.168.45.5:5060;lr;bwgeoproxy>
```

For this scenario to succeed, the access device must support the *Record-Route* mechanism, so that the secondary Application Server can remain in the signaling path.

Following the procedures in *RFC 3261*, the access device copies the received *Record-Route* header into the responses to the INVITE request (F4, F7). The secondary Application Server, however, rewrites its *Record-Route* header entry before forwarding the responses (F5, F8) to the primary Application Server. Specifically, the secondary Application Server adds the *bwpeer* parameter, and may rewrite the IP address in the URI.

In this scenario, the primary Application Server must have a priori knowledge that the access device is unreachable. This knowledge is controlled by an internal *isReachableFromPrimary* indicator that is associated with the device endpoint. The primary Application Server sets this indicator to “false” when it receives an initial INVITE from the secondary Application Server, as in Scenario 1 above. If the indicator is “true”, then the primary Application Server does not route a call to the secondary Application Server, but attempts to route directly to the access device. If the access device is not reachable, then the call fails, as shown in the call flow diagram in *Figure 80*. As shown in the diagram, the primary Application Server does not attempt to route to the secondary Application Server. Note that the SIP error response to the network device is just one possible way for the primary Application Server to handle the call failure.

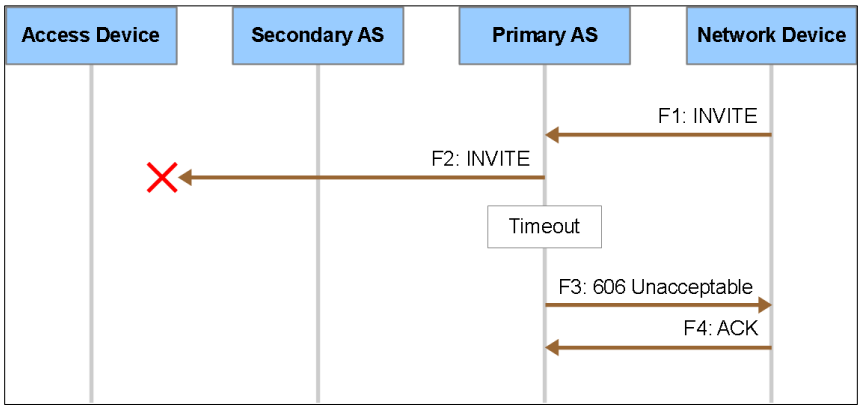

Figure 80 Call Failure Due to Unreachable Access Device

The primary Application Server does not maintain a similar “is reachable” indicator for a network device. Therefore, the primary Application Server always attempts to route an initial INVITE request directly to the network device, and it does not route to the secondary Application Server.

NOTE: In an IMS deployment, the primary Application Server behaves somewhat differently. If the secondary Application Server proxies a new INVITE request to the primary Application Server for an origination, then the primary Application Server does in fact send the outgoing INVITE request to the secondary Application Server, even though the request is sent to a network device. For more information, see the *Cisco BroadWorks AS Mode ISC Interface Specification* [42].

3.52.3 Processing at the Primary Application Server

When the primary Application Server has an initial INVITE request to send to an access device, it may send the INVITE directly to the access device, or it may route it to the secondary Application Server to forward to the access device. These conditions must be met in order for the primary Application Server to route to the secondary Application Server:

- The proxy server behavior must be enabled
- The *isReachableFromPrimary* internal indicator for the destination endpoint must be false
- The secondary Application Server is known to be reachable and able to process SIP messages

The details of the decision logic are described in the *Cisco BroadWorks Redundancy Guide*.

Regardless of whether the primary Application Server sends directly to the access device or to the secondary Application Server, the primary Application Server selects the destination address and transport protocol of the access device, for example, by looking up a contact URI in its location database, resolving a domain name, and so on. If the primary Application Server then routes the INVITE request through the secondary Application Server, it adds a *Route* header entry with the destination address and a *bwdeleteme* parameter. When preparing to send the INVITE request, the primary Application Server also selects the transport protocol (UDP or TCP) to use. If the primary Application Server routes the request through the secondary Application Server, then it uses the selected transport protocol, knowing that the secondary Application Server, acting as a proxy server, will use the same transport protocol to reach the access device.

When the primary Application Server receives a request or response that was proxied by the secondary Application Server, it processes the *X-Broadworks-Source* header (added by the secondary Application Server). The primary Application Server uses this information for various purposes, including checking an access control list, congestion management, and so on.

The primary Application Server can be configured to send OPTIONS requests to access devices to check SIP connectivity. If the proxy server behavior is enabled, and if the *sendSipOptionMessageUponMigration* system parameter is "true", then the primary Application Server sends OPTIONS requests for all device endpoints for which *isReachableFromPrimary* is "false".

3.52.4 Processing at the Secondary Application Server

This section provides details on the behavior of the secondary Application Server acting in the role of a stateless proxy server. For conciseness, the descriptions that follow refer to the secondary Application Server as the "proxy server".

When the proxy server behavior is enabled, the secondary Application Server does not automatically perform as a proxy server when it receives an initial INVITE request. The secondary Application Server may, depending on the specific conditions, decide to handle the call itself. For an explanation of the decision logic, see the *Cisco BroadWorks Redundancy Guide*.

3.52.4.1 Proxy Server Behavior

Max-Forwards Header

The following points describe the proxy server's handling of the *Max-Forwards* header:

- If the received INVITE request does not have a *Max-Forwards* header, then the proxy server adds a *Max-Forwards* header with a value taken from the configured *maxForwardingHops* SIP parameter.
- If the received *Max-Forwards* header has a value higher than *maxForwardingHops*, then the proxy server changes the value to *maxForwardingHops*.
- If the header has a value between 1 and *maxForwardingHops*, inclusive, then the proxy server decrements the value by one.
- If the received *Max-Forwards* header has a value of 0, then the proxy server rejects the request and sends a 483 response.

Via Header

Following the procedures of *RFC 3261*, the proxy server adds a *Via* header entry when it forwards a request. The following points provide additional details:

- If the destination of a request is an access device, then the proxy server adds a new *Via* header entry. To set the *Via* header sent-by field, the proxy server uses:
  - the value of the *bw.sip.accessinterfaceviahost* startup parameter, if it is set
  - otherwise, the public IP address
- If the destination of a request is the primary Application Server or a network device, then the proxy server adds a new *Via* header entry. To set the *Via* header sent-by field, the proxy server uses:
  - the value of the *bw.sip.networkinterfaceviahost* startup parameter, if it is set
  - otherwise, the configured private IP address, if it is configured
  - otherwise, the public IP address
- The proxy server may add a *bwstatelessproxy* parameter to the *Via* header in requests to the primary Application Server. The proxy server uses this information in the response to identify the proxy context.
- The proxy server may add a *prior-port* parameter to the *Via* header when it sends a request over TCP. The proxy server uses this information in the response to determine the prior port.
- A received response must have at least two *Via* header entries. The first entry must be the proxy server's own entry. The second entry must identify the next upstream node. If the proxy server received the response from an access device or a network device, then it forwards the response only if the second *Via* header entry identifies the primary Application Server.

Record-Route Header

For requests that initiate a dialog, the proxy server adds a *Record-Route* header entry. The following points describe the proxy server's handling of the *Record-Route* header:

- When forwarding a request that establishes a dialog, the proxy server adds a *Record-Route* header entry.
- When forwarding a request within a dialog, the proxy server adds a *Record-Route* header entry.
- When forwarding any request outside a dialog, the proxy server adds a *Record-Route* header entry, unless the request is a REGISTER request or a MESSAGE request.
- When forwarding a response that has an expected *Record-Route* header, the proxy server rewrites the *Record-Route* header entry for which it is responsible (this entry should have a *bwgeoproxy* parameter).
- When adding a new *Record-Route* entry to a request, or rewriting the *Record-Route* entry in a response, the proxy server sets the entry's URI based on the setting of a collection of startup parameters. See the table below.
- When adding a new *Record-Route* entry to a request, or rewriting the *Record-Route* entry in a response, the proxy server adds or updates the entry's parameters based on the next hop destination:

- If the next hop is the primary Application Server, the *Record-Route* header entry contains these parameters: *lr*, *bwgeoproxy*, and *bwpeer*. Additionally, if the proxy server received the request from a network device, then it adds a *bwnetwork* parameter.
- If the next hop is an access device, the *Record-Route* header entry contains these parameters: *lr* and *bwgeoproxy*.
- If the next hop is a network device, the *Record-Route* header entry contains these parameters: *lr*, bwgeoproxy, and *bwnetwork*.

Parameter	Description and Alternatives
bw.sip.accessrecordroutehost	This parameter has the *same* meaning as *bw.sip.accessclustercontacthost*. However the proxy server uses it to populate a *Record-Route* entry's host. The default value is "nil". If the value is "nil" and IPv4 is supported, *publicIPAddress* is used if not nil. If "nil", "localhost" is resolved for use. If the value is "nil" and only IPv6 is supported, *publicIPv6Address* is used if not "nil". If "nil", a known IPv6 address is used.
bw.sip.accessrecordrouteincludetcptransport	If this parameter is "true", then the proxy server includes SIP URI transport value TCP when adding or rewriting *Record-Route* entry and sending the request or response over TCP to an access device. The default value is true.
bw.sip.accessrecordrouteincludeudptransport	If this parameter is "true", then the proxy server includes SIP URI transport value UDP when adding or rewriting *Record-Route* entry and sending the request or response over UDP to an access device. The default value is false.
bw.sip.accessrecordrouteport	The proxy server uses this parameter to populate a *Record-Route* entry's port if *bw.sip.accessrecordroutehost* is not "nil". The default value is "nil" which indicates that the port should not be sent. If the value is not "nil", it should be set to the configured SIP *listeningPort* (which defaults to 5060); however this prevents the sender from performing a NAPTR and SRV lookup. If *bw.sip.accessrecordroutehost* is "nil", the configured SIP *listeningPort* is used.
bw.sip.networkrecordroutehost	This parameter has a meaning similar to *bw.sip.accessrecordroutehost*, except it applies to requests or responses sent to network devices. The default value is nil. If the value is "nil" and IPv4 is supported, privateIPAddress is used if not "nil". If privateIPAddress is "nil", publicIPAddress is used if not nil. If "nil", "localhost" is resolved for use. If the value is "nil" and only IPv6 is supported, privateIPv6Address is used if not nil. If privateIPv6Address is "nil", publicIPv6Address is used if not nil. If "nil", a known IPv6 address is used.
bw.sip.networkrecordrouteincludetcptransport	This parameter has a meaning similar to *bw.sip.accessrecordrouteincludetcptransport*, except it applies to requests or responses sent to network devices. The default value is true.
bw.sip.networkrecordrouteincludeudptransport	This has a meaning similar to *bw.sip.accessrecordrouteincludeudptransport*, except it applies to requests or responses sent to network devices. The default value is false.

Parameter	Description and Alternatives
bw.sip.networkrecordrouteport	This parameter has a meaning similar to *bw.sip.accessrecordrouteport*, except it applies to requests or responses sent to network devices. The default value is "nil".
bw.sip.peernetworkrecordroutehost	This parameter a meaning similar to *bw.sip.networkrecordroutehost*, except it applies to requests or responses sent to the peer Application Server. If set, this parameter should be a network side IP-address or hostname corresponding to the peer network interface to this Application Server instance. The default value is "nil". If the value is "nil" and IPv4 is supported, privateIPAddress is used if not "nil". If privateIPAddress is "nil", publicIPAddress is used if not "nil". If "nil", "localhost" is resolved for use. If the value is "nil" and only IPv6 is supported, privateIPv6Address is used if not "nil". If privateIPv6Address is "nil", publicIPv6Address is used if not "nil". If "nil", a known IPv6 address is used.
bw.sip.peernetworkrecordrouteincludetcptransport	This parameter has a meaning similar to *bw.sip.networkrecordrouteincludetcptransport*; except it applies to requests or responses sent to the peer Application Server. The default value is true.
bw.sip.peernetworkrecordrouteincludeudptransport	This parameter has a meaning similar to *bw.sip.networkrecordrouteincludeudptransport*, except it applies to requests or responses sent to the peer Application Server. The default value is false.
bw.sip.peernetworkrecordrouteport	This parameter has a meaning similar to *bw.sip.networkrecordrouteport*, except it applies to requests or responses sent to the peer Application Server. The default value is "nil".

Route Header

When the primary Application Server sends an INVITE request to the proxy server, it adds a pre-loaded *Route* header containing two entries. The first entry identifies the proxy server and has these parameters: *lr*, *bwgeoproxy*, and *bwpeer*. The second entry identifies the next hop destination, as determined by the primary Application Server, and has a *bwdeleteme* parameter. The proxy server removes both *Route* header entries before forwarding the request.

Because the proxy server uses the *Record-Route* mechanism, all in-dialog requests it receives should have a *Route* header as required by *RFC 3261*.

X-BroadWorks-Source Header

The proxy server adds an *X-BroadWorks-Source* header to requests and responses forwarded to the primary Application Server. This header allows the proxy server to convey the source IP-address to the primary Application Server. The primary Application Server can use this address information for access control or other purposes.

Transport Protocol

When the proxy server forwards a request, it sends the outgoing request using the same transport protocol it used to receive the incoming request. For example, if the proxy server receives a request via TCP, then it uses TCP to forward the request. This policy on the proxy server allows the primary Application Server to choose the transport protocol that should be used to reach the remote device.

3.52.4.2 Transaction Tracking

From a SIP perspective, the proxy server is a stateless proxy server. This means that it does not maintain the state necessary to retransmit requests or responses on its own. However, in order to correctly handle a retransmitted request, a CANCEL request, or an ACK request for a non-2xx response, the proxy server does track ongoing transactions for requests received from an access device or network device.

To track transactions more efficiently, the proxy server makes a distinction between short-term transactions and long-term transactions. To start, the proxy server tracks a new transaction as a short-term transaction. It the proxy server receives a 1xx response to an INVITE request, and then it begins to track that transaction as a long-term transaction. The proxy server restarts the timer for a long-term transaction if it receives a retransmitted request, another 1xx response, or a CANCEL request. If it receives a final response for a long-term transaction, then it begins to track that transaction again as a short-term transaction.

The length of time that the proxy server tracks a transaction is controlled by startup parameters:

- *bw.sip.statelessproxyshorttermtrackingseconds* – This parameter sets the minimum time duration for the proxy server to remember a short-term transaction. The range is inclusively 32 to 180 seconds with a default of 32 seconds
- *bw.sip.statelessproxylongtermtrackingseconds* – The parameter sets the minimum time duration for the proxy server to remember a long-term transaction. The range is inclusively 180 to 86,400 seconds with a default of 1800 seconds.
- *bw.sip.statelessproxylongtermtransactionlimit* – This parameter sets the maximum number of long-term transactions the proxy server can track. This parameter can help to protect the system's resources, particularly system memory. When the limit is reached, the proxy server removes the oldest transaction from the tracking list before adding a new one. The range is inclusively 1 to 2,147,483,647 transactions with a default value of 2,147,483,647 transactions.
- *bw.sip.statelessproxyaudittimerseconds* – This parameter controls the frequency of the proxy server's transaction audits, in which it removes expired transactions from the long-term transaction tracking list. The range is inclusively 60 to 86,400 seconds with a default value of "300" seconds.

3.52.5 Peer Monitoring

Before the secondary Application Server routes an initial INVITE request to the primary Application Server (Scenario 1 or Scenario 2, above), the secondary Application Server must know that it has connectivity to the primary Application Server. Likewise, before the primary Application Server routes an initial INVITE request to the secondary Application Server (Scenario 3, above), it must know that it has connectivity to the secondary Application Server. To maintain this connectivity awareness, the primary Application Server and secondary Application Server may be configured to send OPTIONS requests at regular intervals in order to monitor connectivity status.

When SIP connectivity monitoring is enabled, the monitoring Application Server regularly sends an OPTIONS request to its peer Application Server. The interval between OPTIONS requests is configurable via a system parameter. After the monitoring Application Server sends the OPTIONS request, it sets a timer, also controlled by a system parameter, and waits to receive a response. If it receives a response before the timer expires, it considers its peer Application Server to be reachable. Otherwise, when the timer expires, it considers the peer to be unreachable.

Direct monitoring of SIP connectivity is disabled by default. If the Application Server uses the same network interface for SIP messages as for the redundancy link, then the monitoring of the redundancy link is sufficient, and separate monitoring SIP connectivity is unnecessary.

The following configuration values are provided under *System/Redundancy/PeerSipConnectionMonitoring*:

- *enabled* – If this parameter has a "true" value, SIP connectivity monitoring is enabled. The default value is "false".
- *heartbeatIntervalInMsec* – This parameter controls the interval between OPTIONS requests. The default value is "1000" (milliseconds).
- *heartbeatTimeoutInMsec* – This parameter controls the timeout value for the sending Application Server to receive a response to the OPTIONS request. The default value is "5000" (milliseconds).

3.52.6 Syntax

The syntax of the *X-BroadWorks-Source* header is formally defined by the following ABNF.

```
X-BroadWorks-Source = "X-BroadWorks-Source" HCOLON hostport * (SEMI
source-param)
hostport = host [":" port]
host = hostname / IPv4address / IPv6reference
source-param = transport-param / generic-param
transport-param = "transport=" ( "udp" / "tcp" / "sctp" / "tls" / other-
transport)
generic-param = token [ EQUAL gen-value ]
gen-value = token / host / quoted-string
```

The URI parameters used in the *Record-Route* URI and *Route* URI is formally defined by the following ABNF.

```
uri-parameter =/ "bwgeoproxy" / "bwnetwork" / "bwpeer" / "bwdeleteme"
```

The *Via* header parameters are defined by the following ABNF.

```
via-params =/ bwstatelessproxy-param / prior-port-param
bwstatelessproxy-param = "bwstatelessproxy" EQUALS 1*token
prior-port-param = "prior-port" EQUALS port
```

3.52.7 Example Call Flow

The following is a call flow example in which the secondary Application Server acts as a stateless proxy server and relays SIP messages between the primary Application Server and an access device. The diagram shows only the SIP messages between the access device, the secondary Application Server, and the primary Application Server. The messaging shows the secondary Application Server acting as a stateless proxy, which inserts and replaces the proxy's *Record-Route* header entry, adds the *X-BroadWorks-Source* header, and uses the resolved *bwdeleteme Route* header entry. To reduce the size of the example, the message bodies are not shown.

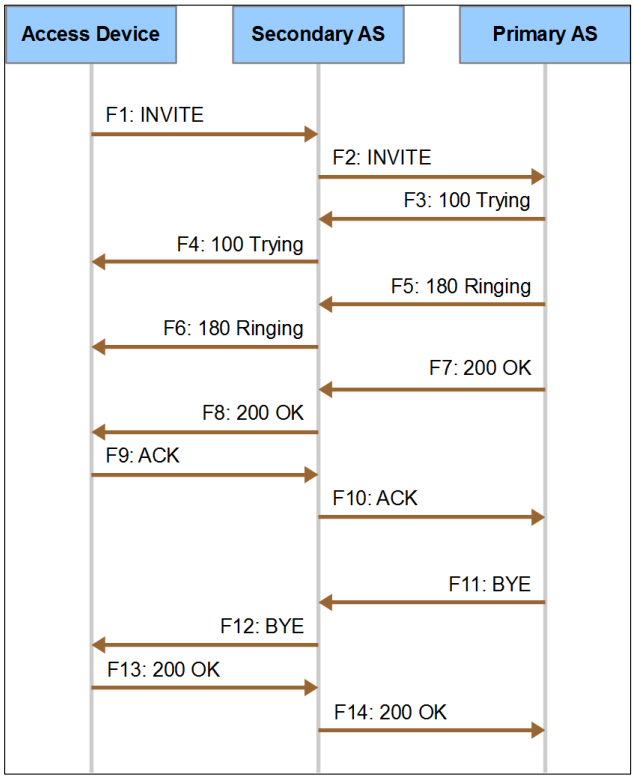

Figure 81 Call Flow Diagram for Secondary Application Server Acting as Proxy Server

F1 INVITE request from access device to secondary Application Server

```
INVITE sip:3015559999@broadsoft.com SIP/2.0
Via: SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
From: <sip:3015558888@broadsoft.com>;tag=unique1
To: <sip:3015559999@broadsoft.com>
Call-ID: 1-5400@10.0.40.40
CSeq: 1 INVITE
Contact: <sip:caller@10.0.40.40>
Max-Forwards: 70
Supported:
Content-Type: application/sdp
Content-Length: 127

(body omitted)
```

F2 INVITE request from secondary Application Server to primary Application Server

```
INVITE sip:3015559999@broadsoft.com SIP/2.0
X-BroadWorks-Source:10.0.40.40
Via:SIP/2.0/UDP 192.168.45.5;branch=z9hG4bKBroadWorksProxy.1qvdmp6.3.805393204,
SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
```

```
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>
Call-ID:1-5400@10.0.40.40
CSeq:1 INVITE
Contact:<sip:caller@10.0.40.40>
Max-Forwards:10
Supported:
Record-Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer>
Content-Type:application/sdp
Content-Length:127

(body omitted)
```

F3 100 (Trying) response from primary Application Server to secondary Application Server

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.45.5;branch=z9hG4bKBroadWorksProxy.1qvdmp6.3.805393204,
SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>
Call-ID:1-5400@10.0.40.40
CSeq:1 INVITE
Content-Length:0
```

F4 100 (Trying) response from secondary Application Server to access device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>
Call-ID:1-5400@10.0.40.40
CSeq:1 INVITE
Content-Length:0
```

F5 180 (Ringing) response from primary Application Server to secondary Application Server

```
SIP/2.0 180 Ringing
Via:SIP/2.0/UDP 192.168.45.5;branch=z9hG4bKBroadWorksProxy.1qvdmp6.3.805393204,
SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID:1-5400@10.0.40.40
CSeq:1 INVITE
Supported:
Contact:<sip:ascluster.broadsoft.com>
Record-Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Content-Length:0
```

F6 180 (Ringing) response from secondary Application Server to access device

```
SIP/2.0 180 Ringing
Via:SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID:1-5400@10.0.40.40
CSeq:1 INVITE
Supported:
Contact:<sip:ascluster.broadsoft.com>
Record-Route:<sip:asclusterrev.broadsoft.com;lr;bwgeoproxy>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Content-Length:0
```

F7 200 (OK) response from primary Application Server to secondary Application Server

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.45.5;branch=z9hG4bKBroadWorksProxy.1qvdmp6.3.805393204,
SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID:1-5400@10.0.40.40
CSeq:1 INVITE
Supported:
Contact:<sip:ascluster.broadsoft.com>
Record-Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Accept:application/media_control+xml,application/sdp
Content-Type:application/sdp
Content-Length:117

(body omitted)
```

F8 200 (OK) response from secondary Application Server to access device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-0
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID:1-5400@10.0.40.40
CSeq:1 INVITE
Supported:
Contact:<sip:ascluster.broadsoft.com>
Record-Route:<sip:asclusterrev.broadsoft.com;lr;bwgeoproxy>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Accept:application/media_control+xml,application/sdp
Content-Type:application/sdp
Content-Length:117

(body omitted)
```

F9 ACK request from access device to secondary Application Server

```
ACK sip:ascluster.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-6
From: <sip:3015558888@broadsoft.com>;tag=unique1
To: <sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID: 1-5400@10.0.40.40
CSeq: 1 ACK
Route:<sip:asclusterrev.broadsoft.com;lr;bwgeoproxy>
Max-Forwards: 70
Content-Length: 0
```

F10 ACK request from secondary Application Server to primary Application Server

```
ACK sip:ascluster.broadsoft.com SIP/2.0
X-BroadWorks-Source:10.0.40.40
Via:SIP/2.0/UDP 192.168.45.5;branch=z9hG4bKBroadWorksProxy.1qvdmp6.3.805393210,
SIP/2.0/UDP 10.0.40.40;branch=z9hG4bK-5400-1-6
From:<sip:3015558888@broadsoft.com>;tag=unique1
To:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID:1-5400@10.0.40.40
CSeq:1 ACK
Max-Forwards:10
Content-Length:0
Record-Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer>
```

F11 BYE request from primary Application Server to secondary Application Server

```
BYE sip:caller@10.0.40.40 SIP/2.0
Via:SIP/2.0/UDP 10.0.55.55;branch=z9hG4bKBroadWorks.1qvdmp6-10.0.40.40V5060-0-
150581070-265255312-1363953280054
From:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
To:<sip:3015558888@broadsoft.com>;tag=unique1
Call-ID:1-5400@10.0.40.40
CSeq:150581070 BYE
Route:<sip:192.168.45.5:5060;lr;bwgeoproxy;bwpeer>,
<sip:10.0.40.40:5060;transport=udp;lr;bwdeleteme>
Max-Forwards:10
Content-Length:0
```

F12 BYE request from secondary Application Server to access device

```
BYE sip:caller@10.0.40.40 SIP/2.0
Via:SIP/2.0/UDP 10.0.45.45;branch=z9hG4bKBroadWorksProxy.1qvdmp6.2.-318229631,
SIP/2.0/UDP 10.0.55.55;received=192.168.55.5;branch=z9hG4bKBroadWorks.1qvdmp6-
10.0.40.40V5060-0-150581070-265255312-1363953280054
From:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
To:<sip:3015558888@broadsoft.com>;tag=unique1
Call-ID:1-5400@10.0.40.40
CSeq:150581070 BYE
Max-Forwards:9
Content-Length:0
Record-Route:<sip:asclusterrev.broadsoft.com;lr;bwgeoproxy>
```

F13 200 (OK) response from access device to secondary Application Server

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.0.45.45;branch=z9hG4bKBroadWorksProxy.1qvdmp6.2.-318229631,
SIP/2.0/UDP 10.0.55.55;received=192.168.55.5;branch=z9hG4bKBroadWorks.1qvdmp6-
10.0.40.40V5060-0-150581070-265255312-1363953280054
From:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
To:<sip:3015558888@broadsoft.com>;tag=unique1
Call-ID:1-5400@10.0.40.40
CSeq:150581070 BYE
Content-Length: 0
```

F14 200 OK response from secondary Application Server to primary Application Server

```
SIP/2.0 200 OK
X-BroadWorks-Source:10.0.40.40
Via:SIP/2.0/UDP 10.0.55.55;received=192.168.55.5;branch=z9hG4bKBroadWorks.1qvdmp6-
10.0.40.40V5060-0-150581070-265255312-1363953280054
From:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
To:<sip:3015558888@broadsoft.com>;tag=unique1
Call-ID:1-5400@10.0.40.40
CSeq:150581070 BYE
Content-Length:0
```

The following is a network side ACK example. The secondary Application Server receives the ACK request and relays it over TCP to the primary Application Server. The example reflects an atypical situation (missing *Via* branch) that causes the secondary Application Server to add the *bwstatelessproxy* parameter to the *Via* header.

ACK request from network device to secondary Application Server

```
ACK sip:asclusternet.broadsoft.com SIP/2.0
Via:SIP/2.0/TCP 10.0.6.60
From:<sip:+13015552222@rfc2543.com;user=phone>;tag=unique1
To:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID:1-5400@192.168.6.60
CSeq:1 ACK
Route:<sip:asclusternetrev.broadsoft.com;lr;bwgeoproxy;bwnetwork>
```

```
Max-Forwards:70
Content-Length:0
```

ACK request from secondary Application Server to primary Application Server

```
ACK sip:asclusternet.broadsoft.com SIP/2.0
X-BroadWorks-Source:192.168.6.60;transport=tcp
Via:SIP/2.0/TCP 192.168.45.5;bwstatelessproxy=4;prior-port=6060, SIP/2.0/TCP
192.168.6.60
From:<sip:+13015552222@rfc2543.com;user=phone>;tag=unique1
To:<sip:3015559999@broadsoft.com>;tag=265255312-1363953280054
Call-ID:1-5400@192.168.6.60
CSeq:1 ACK
Max-Forwards:10
Content-Length:0
Record-Route:<sip:192.168.45.5:5060;transport=tcp;lr;bwgeoproxy;bwpeer;bwnetwork>
```

3.53 Preconditions Framework (RFC 3312)

Reference Documents:

- *RFC 3312: Integration of Resource Management and Session Initiation Protocol (SIP)*, October 2002

3.53.1 Cisco BroadWorks Support for Preconditions

Cisco BroadWorks support for preconditions is controlled by the SIP system parameter *suppressRFC3312Preconditions*, which can take one of these values: “always”, “never”, and “suppressIfSingleDialog”. The default value is “always”.

When *suppressRFC3312Preconditions* is set to “always”, Cisco BroadWorks suppresses preconditions. If the received SDP contains preconditions attributes, Cisco BroadWorks removes those attributes before relaying the SDP to the terminating endpoint. More specifically, Cisco BroadWorks removes the following SDP attributes:

- a=curr
- a=des
- a=conf

If the incoming INVITE request has a *Supported* header with “preconditions”, then Cisco BroadWorks removes “preconditions” from the *Supported* header in the outgoing INVITE request. If the incoming INVITE request has a *Require* header with “preconditions”, then Cisco BroadWorks sends a *420 Unsupported Extension* response with “preconditions” in the *Unsupported* header.

When *suppressRFC3312Preconditions* is set to “never”, Cisco BroadWorks supports preconditions. If the received SDP contains preconditions attributes, Cisco BroadWorks copies those attributes into the sent SDP. If the incoming INVITE request has a *Supported* header with “preconditions”, then Cisco BroadWorks sends the *Supported* header with “preconditions” in the outgoing INVITE request. If the incoming INVITE request has a *Require* header with “preconditions”, then Cisco BroadWorks sends the *Require* header with “preconditions” in the outgoing INVITE request.

When *suppressRFC3312Preconditions* is set to “suppressIfSingleDialog”, then Cisco BroadWorks behavior depends on the operating mode of the originating session. If Cisco BroadWorks operates in single dialog mode, then Cisco BroadWorks suppresses preconditions. Otherwise, if Cisco BroadWorks operates in multiple dialog mode, then it supports preconditions. Single dialog mode and multiple dialog mode are described in section *3.14 SIP Forking*.

When Cisco BroadWorks connects the originating endpoint to the Media Server, whether before answer for a service such as Custom Ringback or after answer for announcements or IVR, it can suppress or support preconditions. When Cisco BroadWorks is configured to suppress preconditions, it skips preconditions negotiation. When Cisco BroadWorks is configured to support preconditions, it negotiates preconditions with the originating endpoint. Because the Media Server does not need to allocate resources for preconditions, the Application Server negotiates preconditions with the originating endpoint on behalf of the Media Server. Before answer, the Application Server waits until preconditions negotiation is complete before it starts Media Server streaming. Likewise, the Application Server does not answer the call until preconditions negotiation is complete.

3.53.2 Interactions Cisco BroadWorks Forking Services

If Cisco BroadWorks operates in a mode where it consumes provisional responses from secondary device endpoints, then it permits preconditions negotiation only between the originating endpoint and the primary device endpoint. In this mode, Cisco BroadWorks suppresses preconditions negotiation with the secondary endpoints. If the incoming INVITE request has a *Require* header with “preconditions”, and if the called user requires forking to secondary endpoints, then Cisco BroadWorks rejects the INVITE request with a 580 (Precondition Failure) response.

On the other hand, if Cisco BroadWorks operates in a mode where it relays provisional responses from secondary endpoints, then it permits preconditions negotiation with secondary endpoints as well as the primary device endpoint. Preconditions negotiation in this mode is summarized in the following points:

- If the INVITE request from the originating endpoint contains a *Supported* header with “preconditions”, then Cisco BroadWorks also sends *Supported* with “preconditions” to the secondary endpoints.
- If the INVITE request from the originating endpoint contains a *Require* header with “preconditions”, then Cisco BroadWorks also sends *Require* with “preconditions” to the secondary endpoints.
- Cisco BroadWorks relays the preconditions attributes in SDP between the originating endpoint and the secondary terminating endpoints.

3.54 User Agent Capabilities (RFC 3840)

Cisco BroadWorks generally does not support this functionality. However, Cisco BroadWorks does provide an option to support the *sip.video* media feature tag.

3.54.1 Support for sip.video Media Feature Tag

If the SIP system parameter *transmitIR94DeviceVideoCapability* is set to "true", then Cisco BroadWorks supports the *sip.video* media feature tag. When this support is enabled and Cisco BroadWorks receives an incoming INVITE request that contains the "video" tag[4], Cisco BroadWorks stores this information and may send the "video" tag in outgoing INVITE requests on behalf of the remote user agent client (UAC). Similarly, when Cisco BroadWorks receives a 200 response (to an INVITE request) that contains the "video" tag, Cisco BroadWorks stores that information and may send the "video" tag in the outgoing 200 response on behalf of the remote user agent server (UAS).

Cisco BroadWorks sends the "video" tag in outgoing requests and responses only if it is configured to support video on the destination endpoint. For an access device, this means the device profile type of the device endpoint must have the *Video Capable* option enabled. For a network device, the system parameter *networkSupportVideo* must be enabled.

When support for the *sip.video* media feature tag is enabled, Cisco BroadWorks support generally follows the requirements of *RFC 3840* [68][68] and GSMA IR.94 [69]. Thus, Cisco BroadWorks supports the "video" tag in SIP requests and responses that initiate a dialog (initial INVITE requests and their 200 responses) as well as target refresh requests and responses (re-INVITE requests, UPDATE requests, and their 200 responses).

Because Cisco BroadWorks stores information about the received "video" tag, Cisco BroadWorks is able to support this tag in a variety of call scenarios. For example, if a caller sends the "video" tag to indicate support for video media and the callee transfers the call after answer, then Cisco BroadWorks may send the "video" tag to the transfer-to party on behalf of the caller.

Note also the following additional remarks:

- Cisco BroadWorks may send a "video" tag on behalf of a Cisco BroadWorks user only if it receives a "video" tag in a SIP message. Thus, there is no device configuration option to cause Cisco BroadWorks to generate a "video" tag on behalf of a Cisco BroadWorks user.
- If Cisco BroadWorks receives a "video" tag from a Cisco BroadWorks user's device, then that tag does not temporarily override the device configuration to indicate that the device endpoint supports video.

If *transmitIR94DeviceVideoCapability* is set to "false", then Cisco BroadWorks does not send the "video" media feature tag.

For a new installation, the default value for *transmitIR94DeviceVideoCapability* is "true".

[4] The registered name of the media feature tag is *sip.video*. However, when the tag appears as a parameter in the *Contact* header, the "sip." prefix is omitted. Therefore, this document refers to the tag as the "video" tag in the context of a SIP message.

4 Call Flows

This section contains call flows of various scenarios that must be supported by a network device. The flows show the message flow and a detailed example of each scenario. The flows only include the minimal amount of *SIP* headers and messages required for a network device to interwork with Cisco BroadWorks. The following scenarios are depicted:

- User to network call
- Network to user call using SIP URI addressing
- Network to user call using Tel URI addressing
- User release
- Network release
- User to network call with privacy requested (*RFC 3323/3325*)
- Network to user call with privacy requested (*RFC 3323/3325*)
- User to network call with privacy requested (*draft-ietf-sip-privacy-03*)
- Network to user call with privacy requested (*draft-ietf-sip-privacy-03*)
- User to network call with redirection (diversion), unconditional call forwarding
- User places call on hold
- User retrieves held call
- User-initiated media request (used in blind transfer, transfer, conferencing, and so on)
- User to network with calling party category/originating line information
- Network to Cisco BroadWorks Network Server call redirection
- User to network with equal access via Cisco BroadWorks Network Server
- User to network with originating trunk group via Cisco BroadWorks Network Server
- Network to Cisco BroadWorks subscription (generic-event event package example)
- Cisco BroadWorks to network subscription (calling-name event package example)
- Network to Cisco BroadWorks message waiting indication
- Network to Cisco BroadWorks instant message

Each flow shows an annotated example of a user to network call or network to user call, where the Cisco BroadWorks Application Server is acting on behalf of the user.

4.1 User to Network Call

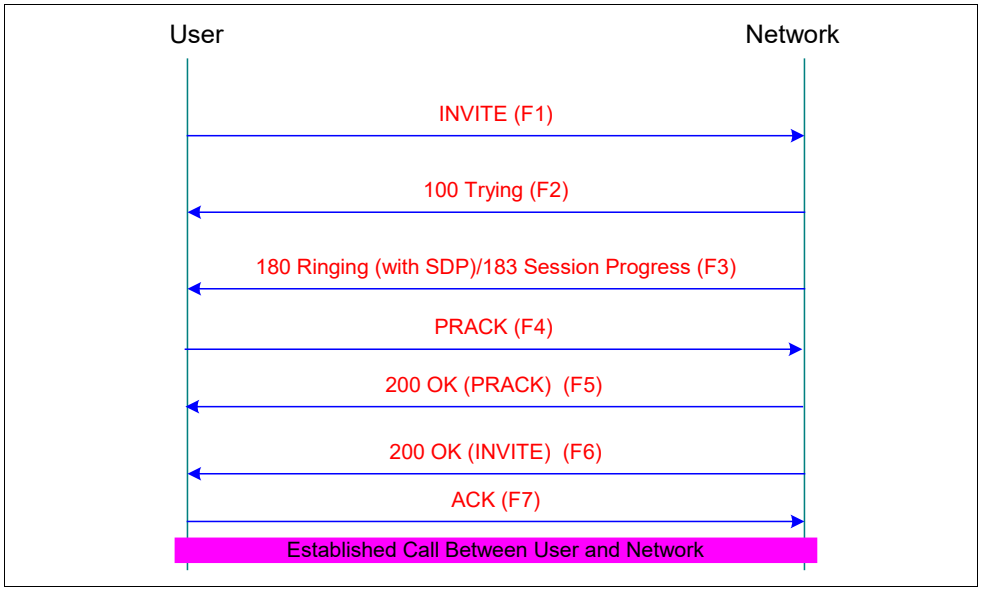

Figure 82 User to Network Call

4.1.1 F1 – INVITE: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Contact:<sip:applicationserver.broadsoft.com>
P-Asserted-Identity:”Joe Smith”<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:none
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.1.2 F2 – 100 Trying: Network → User

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Content-Length: 0
```

4.1.3 F3 – 180 Ringing (with SDP)/183 Session Progress: Network → User

```
SIP/2.0 183 Session Progress
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Content-Type: application/sdp
Session: Media
Require: 100rel
RSeq: 6094
Content-Length: 135

v=0
o=- 8106 395 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20982 RTP/AVP 0
```

4.1.4 F4 – PRACK: User → Network

```
PRACK sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815250 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815250 PRACK
RAck:6094 1000815249 INVITE
Max-Forwards:10
Content-Length: 0
```

4.1.5 F5 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815250 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815250 PRACK
Content-Length: 0
```

4.1.6 F6 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
Content-Type: application/sdp
Content-Length: 135

v=0
o=- 8106 395 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20982 RTP/AVP 0
```

4.1.7 F7 – ACK: User → Network

```
ACK sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-
1000815249A1646372935-985549141285From:"Joe
Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 ACK
Contact:<sip:applicationserver.broadsoft.com>
Max-Forwards:10
Content-Length:0
```

4.2 Network to User Call Using SIP URI Addressing

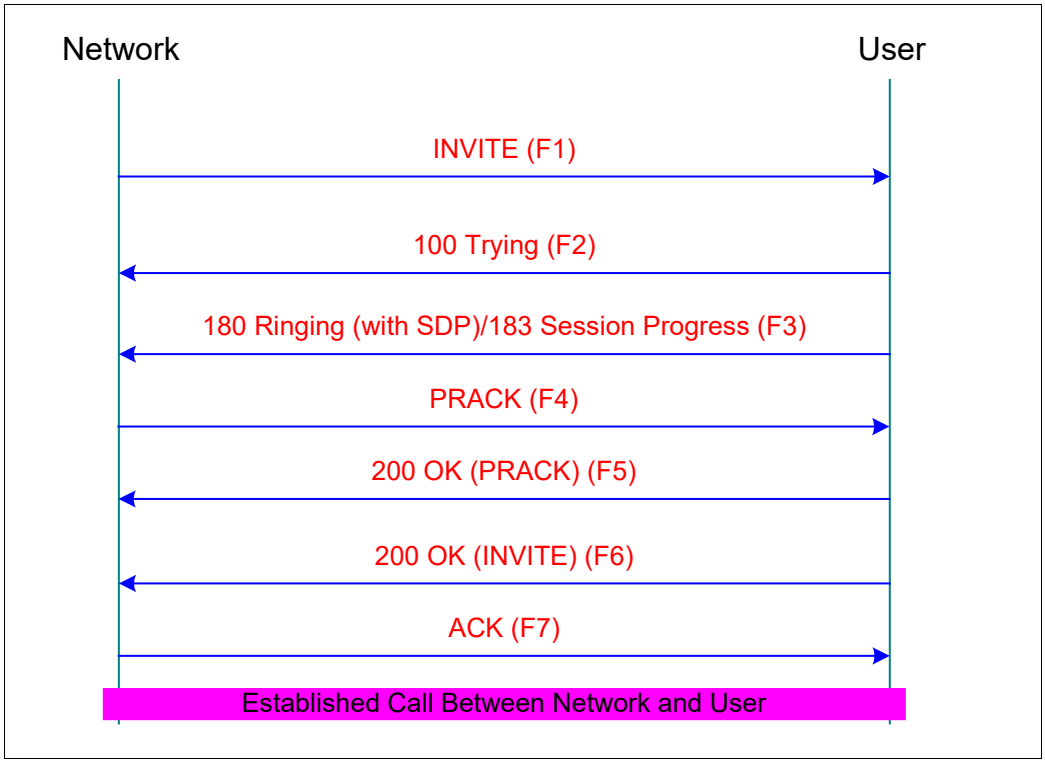

Figure 83 Network to User Call Using SIP URI Addressing

4.2.1 F1 – INVITE: Network → User

```
INVITE sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Contact: <sip:+13015400460@networkdevice.com;user=phone>
Max-Forwards:10
Supported:100rel,timer
Content-Type: application/sdp
Content-Length: 136

v=0
o=- 4851 3460 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20268 RTP/AVP 0
```

4.2.2 F2 – 100 Trying: User → Network

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Content-Length:0
```

4.2.3 F3 – 180 Ringing (with SDP)/183 Session Progress: User → Network

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Content-Type:application/sdp
Require: 100rel
RSeq: 1234
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.2.4 F4 – PRACK: Network → User

```
PRACK sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 102 PRACK
RAck:1234 101 INVITE
Max-Forwards:10
Content-Length: 0
```

4.2.5 F5 – 200 OK: User → Network

```
SIP/2.0 200 OK
 Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq:102 PRACK
Content-Length: 0
```

4.2.6 F6 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Contact:<sip:applicationserver.broadsoft.com>
Content-Type:application/sdp
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
```

```
a=rtpmap:0 PCMU/8000
```

4.2.7 F7 – ACK: Network → User

```
ACK sip:applicationserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 ACK
Max-Forwards:10
Content-Length: 0
```

4.3 Network to User Call Using Tel URI Addressing

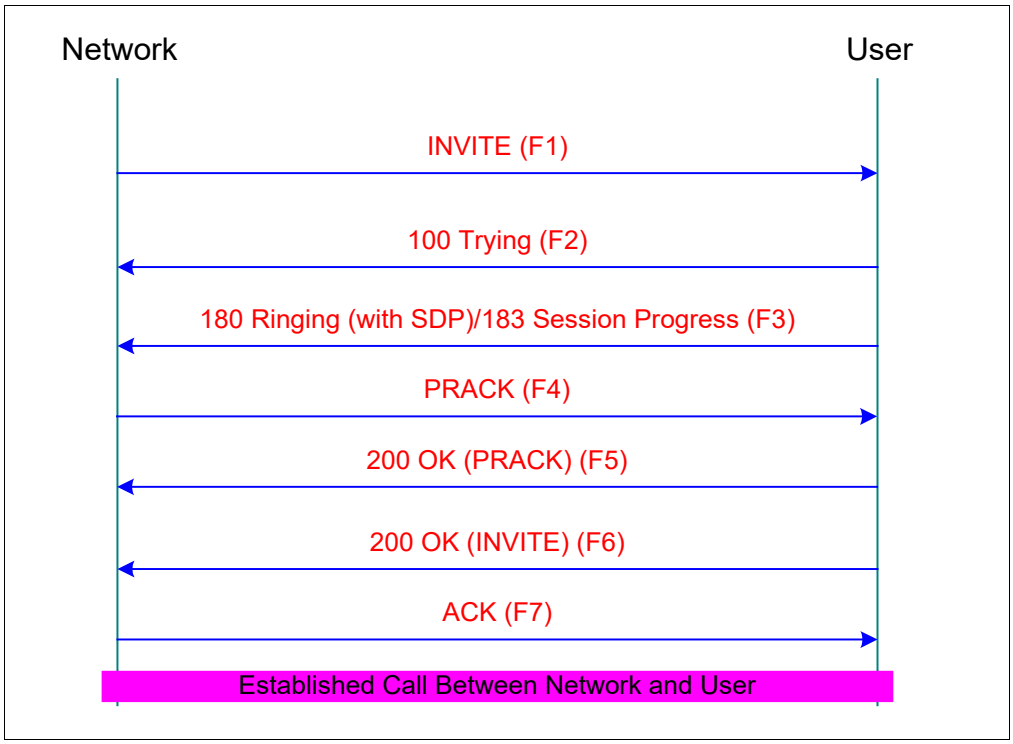

Figure 84 Network to User Call Using Tel URI Addressing

4.3.1 F1 – INVITE: Network → User

```
INVITE tel:+12403645111 SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: < tel:+12403645111>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Contact: <sip:+13015400460@networkdevice.com;user=phone>
Max-Forwards:10
Supported:100rel,timer
Content-Type: application/sdp
Content-Length: 136

v=0
o=- 4851 3460 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
```

```
m=audio 20268 RTP/AVP 0
```

4.3.2 F2 – 100 Trying: User → Network

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: < tel:+12403645111>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Content-Length:0
```

4.3.3 F3 – 180 Ringing (with SDP)/183 Session Progress: User → Network

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: < tel+12403645111>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Content-Type:application/sdp
Require: 100rel
RSeq: 1234
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.3.4 F4 – PRACK: Network → User

```
PRACK tel:2403645111 SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: < tel:+12403645111>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 102 PRACK
RAck:1234 101 INVITE
Max-Forwards:10
Content-Length: 0
```

4.3.5 F5 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: < tel:+12403645111>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq:102 PRACK
Content-Length: 0
```

4.3.6 F6 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: < tel:+12403645111>;tag=1954344904-985549204867
```

```
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Contact:<sip:applicationserver.broadsoft.com>
Content-Type:application/sdp
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.3.7 F7 – ACK: Network → User

```
ACK sip:applicationserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: < tel:+12403645111>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 ACK
Max-Forwards:10
Content-Length: 0
```

4.4 User to Network Call, User Releases Call

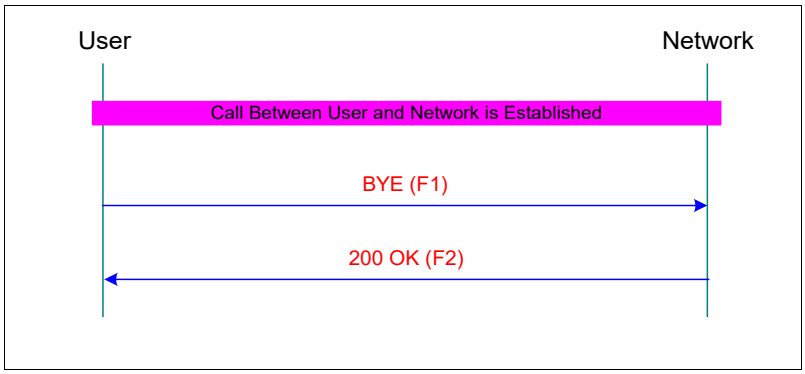

Figure 85 User to Network Call, User Releases Call

4.4.1 F1 – BYE: User → Network

```
BYE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815868 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815868 BYE
Max-Forwards:10
Content-Length:0
```

4.4.2 F2 – 200 OK: Network → User

```
SIP/2.0 200 OK
```

```
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815868 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815868 BYE
Content-Length:0
```

4.5 Network to User Call, Network Releases Call

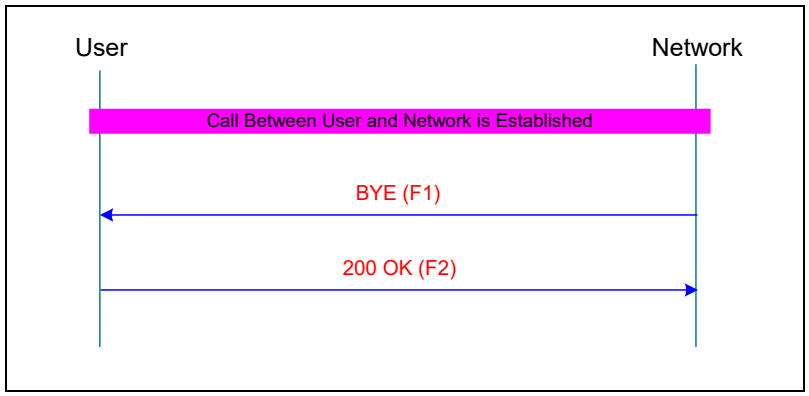

Figure 86 Network to User Call, Network Releases Call

4.5.1 F1 – BYE: Network → User

```
BYE sip:applicationserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:2403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815868 BYE
Max-Forwards:10
Content-Length:0
```

4.5.2 F2 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:2403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815868 BYE
Content-Length:0
```

4.6 User to Network Call with Privacy Requested (RFC 3323/3325)

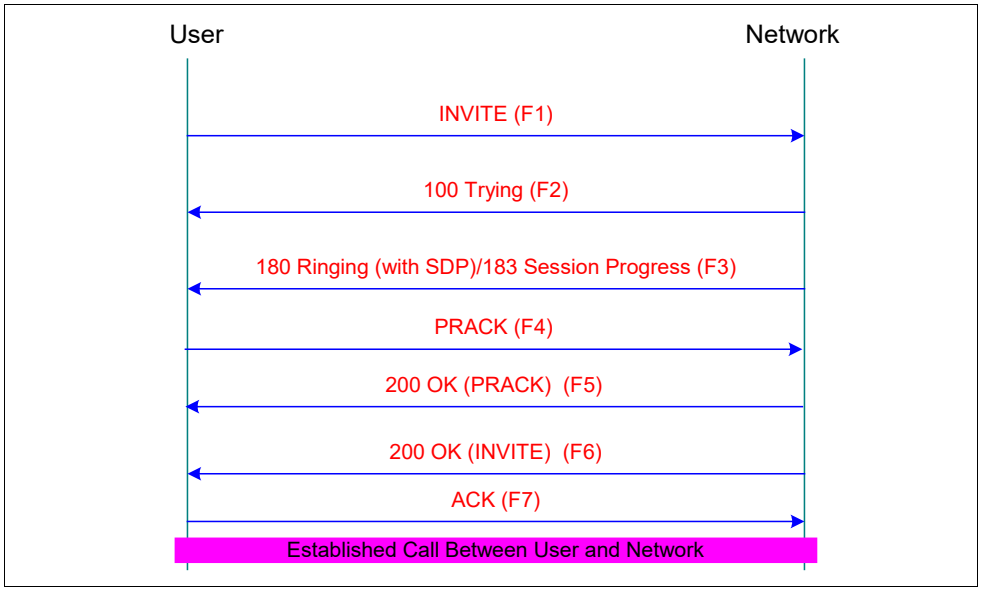

Figure 87 User to Network Call with Privacy Requested (RFC 3323/3325)

4.6.1 F1 – Invite: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-
1646372935-985549141285
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Contact:<sip:applicationserver.broadsoft.com>
P-Asserted-Identity: "Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:user;critical;id
 Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.6.2 F2 – 100 Trying: Network → User

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-
1646372935-985549141285
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID: BW143901028525030101313124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Content-Length: 0
```

4.6.3 F3 – 180 Ringing (with SDP)/183 Session Progress: Network → User

```
SIP/2.0 183 Session Progress
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-
1646372935-985549141285
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW143901028525030101313124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Content-Type: application/sdp
Session: Media
Require: 100rel
RSeq: 6094
Content-Length: 135

v=0
o=- 8106 395 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20982 RTP/AVP 0
```

4.6.4 F4 – PRACK: User → Network

```
PRACK sip:+13015400460@networkdevice.com SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815250-
1646372935-985549141285
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID:BW143901028525030101313124125576944@applicationserver.broadworks.com
CSeq:1000815250 PRACK
RAck:6094 1000815249 INVITE
Max-Forwards:10
Content-Length: 0
```

4.6.5 F5 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815250-
1646372935-985549141285
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID:BW143901028525030101313124125576944@applicationserver.broadworks.com
CSeq:1000815250 PRACK
Content-Length: 0
```

4.6.6 F6 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-
1646372935-985549141285
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW143901028525030101312412557694@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
CSeq: 1000815249 INVITE
Content-Type: application/sdp
Content-Length: 135

v=0
o=- 8106 395 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20982 RTP/AVP 0
```

4.6.7 F7 – ACK: User → Network

```
ACK sip:+13015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-
1000815249A1646372935-985549141285
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW143901028525030101312412557694@applicationserver.broadworks.com
CSeq: 1000815249 ACK
Contact:<sip:applicationserver.broadsoft.com>
Max-Forwards:10
Content-Length:0
```

4.7 Network to User Call with Privacy Requested (RFC 3323/3325)

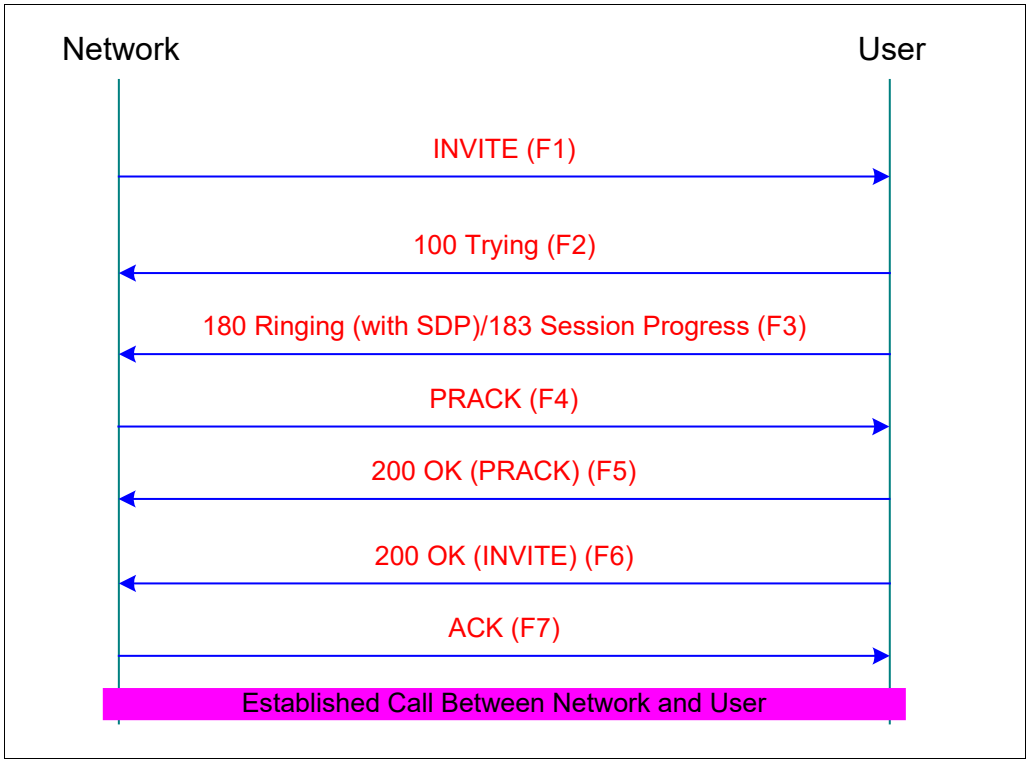

Figure 88 Network to User Call with Privacy Requested (RFC 3323/3325)

4.7.1 F1 – INVITE: Network → User

```
INVITE sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
P-Asserted-Identity: "Bob Smith"<sip:+13015400460@networkdevice.com;user=phone>
Privacy:user;critical;id
Max-Forwards:10
Supported:100rel,timer
Content-Type: application/sdp
Content-Length: 136

v=0
o=- 4851 3460 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20268 RTP/AVP 0
```

4.7.2 F2 – 100 Trying: User → Network

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  10.10.180.72:54938
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
```

```
Content-Length:0
```

4.7.3 F3 – 180 Ringing (with SDP)/183 Session Progress: User → Network

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  10.10.180.72:54938
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Content-Type:application/sdp
Require: 100rel
RSeq: 1234
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.7.4 F4 – PRACK: Network → User

```
PRACK sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 102 PRACK
RAck:1234 101 INVITE
Max-Forwards:10
Content-Length: 0
```

4.7.5 F5 – 200 OK: User → Network

```
SIP/2.0 200 OK
 Via: SIP/2.0/UDP  10.10.180.72:54938
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq:102 PRACK
Content-Length: 0
```

4.7.6 F6 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Contact:<sip:applicationserver.broadsoft.com>
Content-Type:application/sdp
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
```

```
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.7.7 F7 – ACK: Network → User

```
ACK sip:applicationserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 ACK
Max-Forwards:10
Content-Length: 0
```

4.8 User to Network Call with Privacy Requested (draft-ietf-sip-privacy-03)

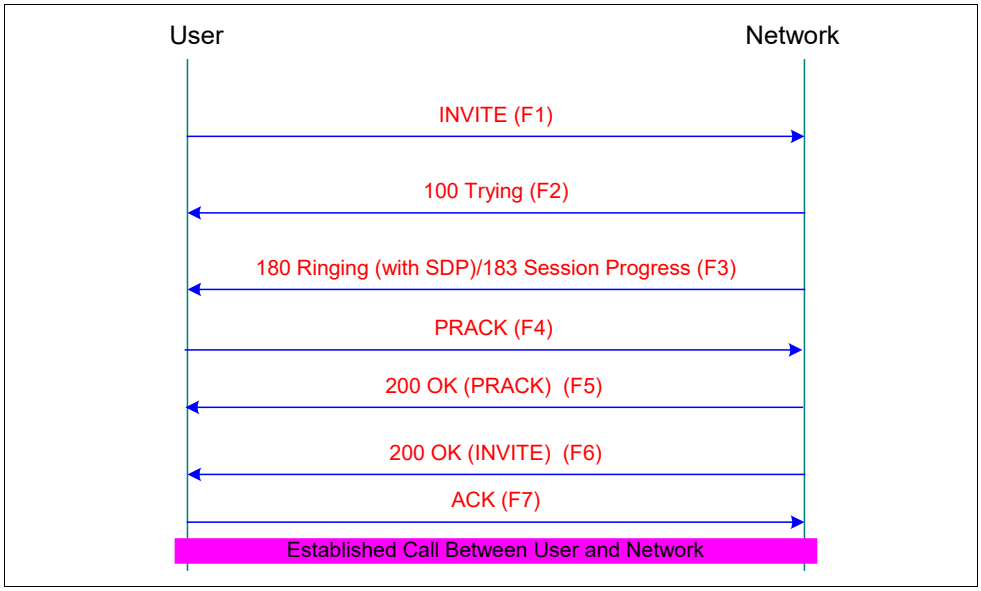

Figure 89 User to Network Call with Privacy Requested (draft-ietf-sip-privacy-03)

4.8.1 F1 – Invite: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-
1646372935-985549141285
From:"anonymous"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Contact:<sip:applicationserver.broadsoft.com>
RPID-Privacy:party=calling;id-type=subscriber;privacy=full
Remote-Party-ID:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com>;party=calling;id-
type=subscriber;privacy=full;screen=yes
Proxy-Require:privacy
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
```

```
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.8.2 F2 – 100 Trying: Network → User

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-1646372935-985549141285
From:" anonymous "<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Content-Length: 0
```

4.8.3 F3 – 180 Ringing (with SDP)/183 Session Progress: Network → User

```
SIP/2.0 183 Session Progress
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-1646372935-985549141285
From:" anonymous "<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Remote-Party-ID:"Bob Smith"<sip:+13015400460@networkdevice.com >;party=called;id-type=subscriber;privacy=off;screen=yes
Content-Type: application/sdp
Session: Media
Require: 100rel
RSeq: 6094
Content-Length: 135

v=0
o=- 8106 395 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20982 RTP/AVP 0
```

4.8.4 F4 – PRACK: User → Network

```
PRACK sip:+13015400460@networkdevice.com SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815250-1646372935-985549141285
From:"anonymous"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815250 PRACK
RAck:6094 1000815249 INVITE
Max-Forwards:10
Content-Length: 0
```

4.8.5 F5 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815250-1646372935-985549141285
From:"anonymous"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815250 PRACK
Content-Length: 0
```

4.8.6 F6 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-1646372935-985549141285
From:" anonymous "<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
RPID-Privacy:party=called;id-type=subscriber;privacy=off
Remote-Party-ID:"Bob Smith"<sip:+13015400460@networkdevice.com >;party=called;id-type=subscriber;privacy=off;screen=yes
Content-Type: application/sdp
Content-Length: 135

v=0
o=- 8106 395 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20982 RTP/AVP 0
```

4.8.7 F7 – ACK: User → Network

```
ACK sip:+13015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249A1646372935-985549141285
From:" anonymous "<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1
Call-ID: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 ACK
Contact:<sip:applicationserver.broadsoft.com>
Max-Forwards:10
Content-Length:0
```

4.9 Network to User Call with Privacy Requested (draft-ietf-sip-privacy-03)

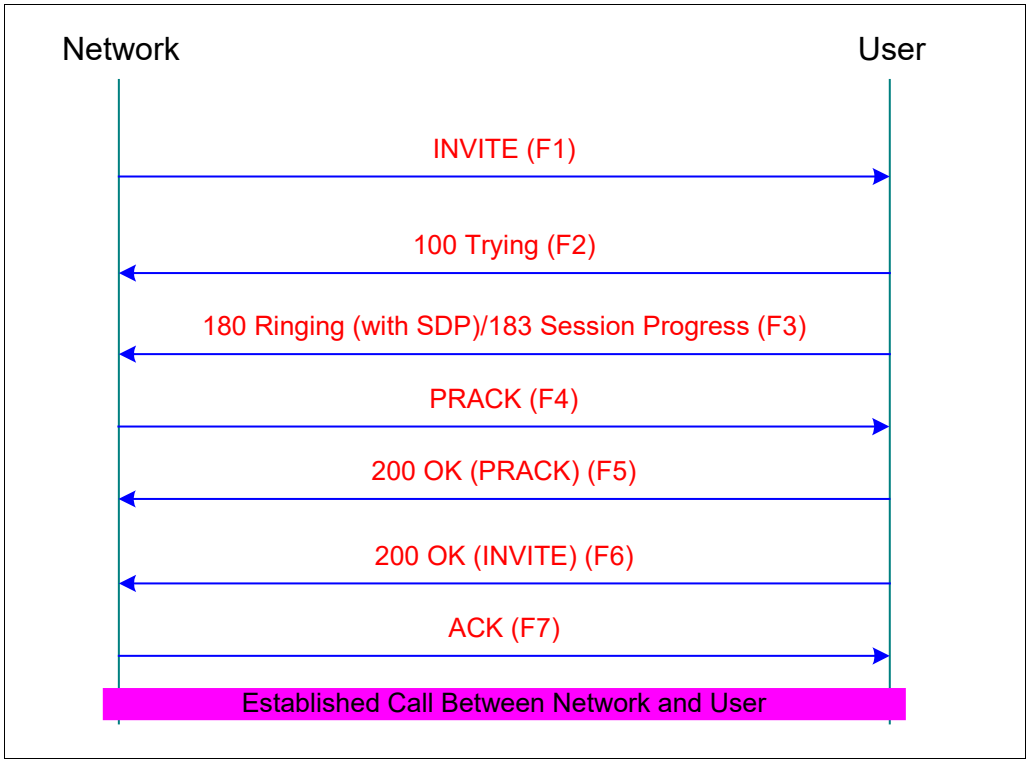

Figure 90 Network to User Call with Privacy Requested (draft-ietf-sip-privacy-03)

4.9.1 F1 – INVITE: Network → User

```
INVITE sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "anonymous" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
RPID-Privacy:party=calling;id-type=subscriber;privacy=full
Remote-Party-ID:"Bob Smith"<sip:+13015400460@networkdevice.com >;party=calling;id-
type=subscriber;privacy=full;screen=yes
Proxy-Require:privacy
Max-Forwards:10
Supported:100rel,timer
Content-Type: application/sdp
Content-Length: 136

v=0
o=- 4851 3460 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20268 RTP/AVP 0
```

4.9.2 F2 – 100 Trying: User → Network

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  10.10.180.72:54938
From: " anonymous" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Content-Length:0
```

4.9.3 F3 – 180 Ringing (with SDP)/183 Session Progress: User → Network

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  10.10.180.72:54938
From: " anonymous " <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Remote-Party-ID:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com>;party=called;id-
type=subscriber;privacy=off;screen=yes
Content-Type:application/sdp
Require: 100rel
RSeq: 1234
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.9.4 F4 – PRACK: Network → User

```
PRACK sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "anonymous" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 102 PRACK
RAck:1234 101 INVITE
Max-Forwards:10
Content-Length: 0
```

4.9.5 F5 – 200 OK: User → Network

```
SIP/2.0 200 OK
 Via: SIP/2.0/UDP  10.10.180.72:54938
From: "anonymous" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq:102 PRACK
Content-Length: 0
```

4.9.6 F6 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: " anonymous " <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Contact:<sip:applicationserver.broadsoft.com>
RPID-Privacy:party=called;id-type=subscriber;privacy=off
Remote-Party-ID:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com>;party=called;id-
type=subscriber;privacy=off;screen=yes
Supported:100rel, timer
Accept::application/sdp
Content-Type:application/sdp
Content-Length:102

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.9.7 F7 – ACK: Network → User

```
ACK sip:applicationserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: " anonymous " <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 ACK
Max-Forwards:10
Content-Length: 0
```

4.10 User to Network Call with Redirection (Diversion), Unconditional Call Forwarding

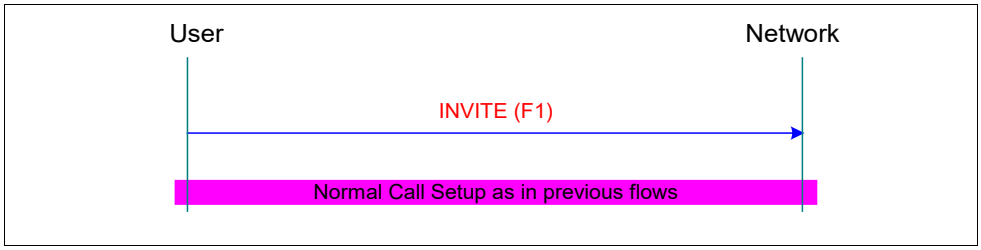

Figure 91 User to Network Call with Redirection (Diversion), Unconditional Call Forwarding

4.10.1 F1 – INVITE: User → Network

```
INVITE sip:2406329348@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249-
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:2406329348@networkdevice.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Contact:<sip:applicationserver.broadsoft.com>
P-Asserted-Identity:”Joe Smith”<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
```

```
Privacy:none
Diversion:"Bob
Smith"<sip:+13015400460@applicationserver.broadsoft.com>;reason=unconditional;counter=1;privacy=off
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.11 User Places Call on Hold

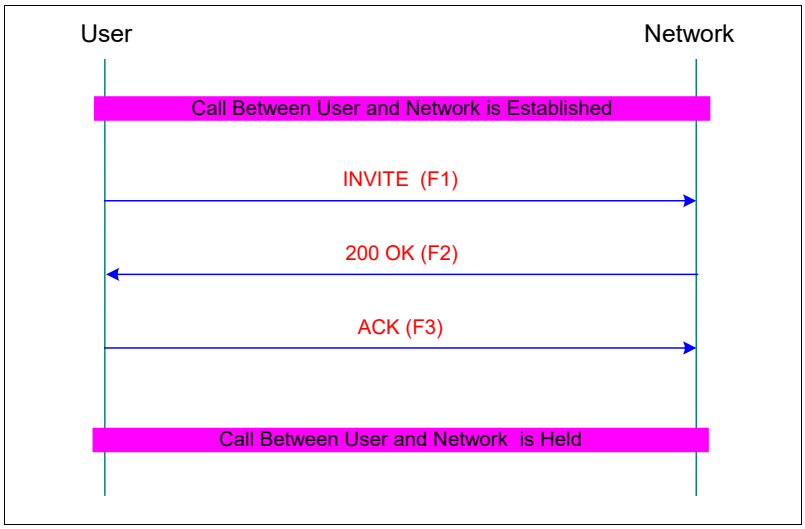

Figure 92 User Places Call on Hold

4.11.1 F1 – Re-INVITE: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956-
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 INVITE
Contact:<sip:applicationserver.broadsoft.com>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:101

v=0
o=BroadWorks 3 2 IN IP4 192.168.5.215
s=-
c=IN IP4 0.0.0.0
t=0 0
```

```
m=audio 16864 RTP/AVP 0
a=inactive
```

4.11.2 F2 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956-1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
Content-Type: application/sdp
Content-Length:102

v=0
o=- 566 5805 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20950 RTP/AVP 0
a=inactive
```

4.11.3 F3 – ACK: User → Network

```
ACK sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956A1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 ACK
Contact:<sip:applicationserver.broadsoft.com>
Max-Forwards:10
Content-Length:0
```

4.12 User Retrieves Held Call

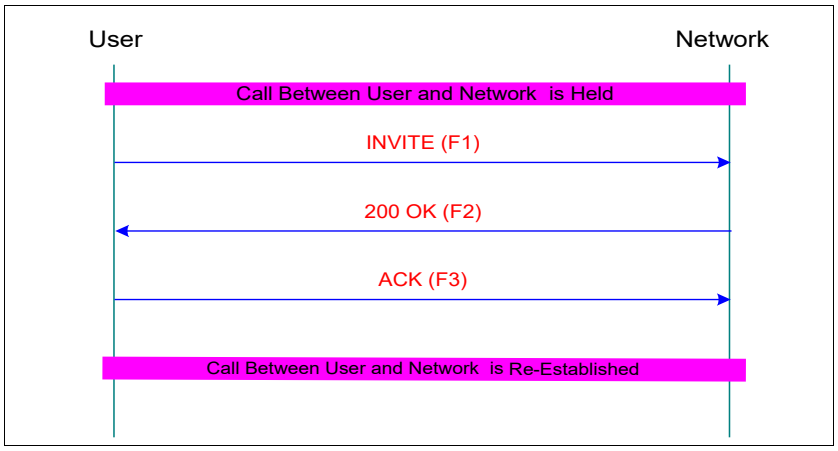

Figure 93 User Retrieves Held Call

4.12.1 F1 – Re-Invite: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956-1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 INVITE
Contact:<sip:applicationserver.broadsoft.com>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:101

v=0
o=BroadWorks 3 3 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.12.2 F2 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956-1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
Content-Type: application/sdp
Content-Length:102

v=0
o=- 566 5806 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20950 RTP/AVP 0
a=sendrecv
```

4.12.3 F3 – ACK: User → Network

```
ACK sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956A1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 ACK
Contact:<sip:applicationserver.broadsoft.com>
Max-Forwards:10
Content-Length:0
```

4.13 User-initiated Media Request

A user-initiated media request could occur for a variety of call control scenarios including blind transfer, consultation transfer, conferencing, call park, and so on.

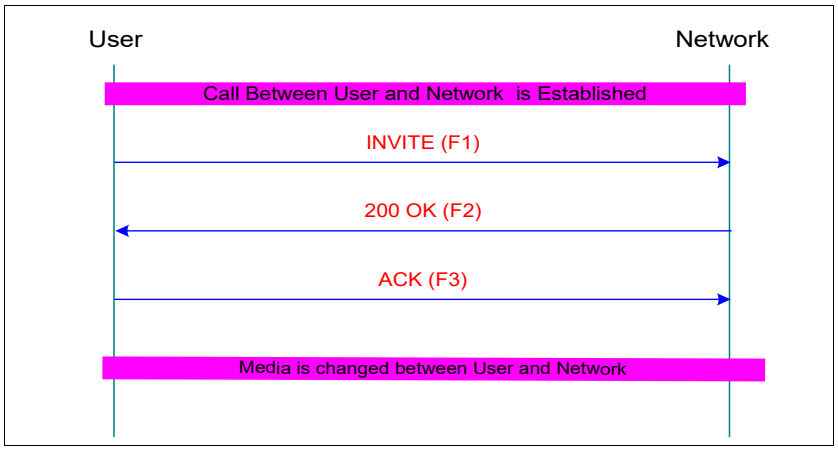

Figure 94 User-initiated Media Request

4.13.1 F1 – Re-Invite: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956-
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 INVITE
Contact:<sip:applicationserver.broadsoft.com>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Length:0
```

4.13.2 F2 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815956-
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 INVITE
Contact: <sip:3015400460@networkdevice.com;user=phone>
Content-Type: application/sdp
Content-Length:102

v=0
o=- 566 5806 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20950 RTP/AVP 0
a=sendrecv
```

4.13.3 F3 – ACK: User → Network

```
ACK sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-
1000815956A1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>;tag=1954344904-985549204867
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815956 ACK
Contact:<sip:applicationserver.broadsoft.com>
Max-Forwards:10
Content-Type:application/sdp
Content-Length:101

v=0
o=BroadWorks 3 2 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.14 User to Network with Calling Party Category/Originating Line Information

The flow shown in *Figure 95* is a user to network INVITE, which contains the *calling party category* or *originating line information* parameters. Note that the rest of the flow is identical to the user to network flow previously described.

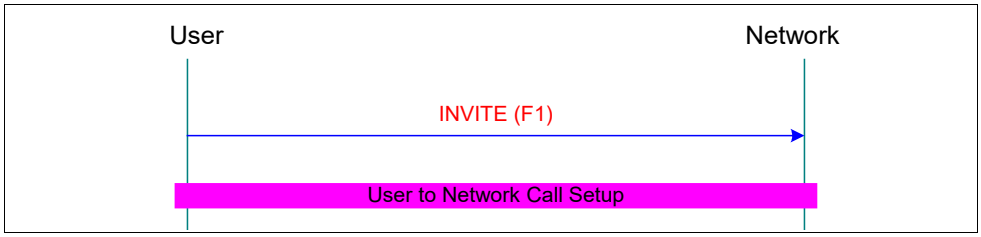

Figure 95 User to Network with Calling Party Category/Originating Line Information

4.14.1 Calling Party Category (CPC Parameter)

4.14.1.1 F1 – INVITE: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe
Smith"<sip:+12403645111;cpc=payphone@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Contact:<sip:applicationserver.broadsoft.com>
P-Asserted-Identity:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:none
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
```

```
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.14.2 Originating Line Information (isup-oli Parameter)

4.14.2.1 F1 – INVITE: User → Network

```
INVITE sip:3015400460@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone;isup-
oli=70>;tag=1646372935-985549141285
To:<sip:3015400460@networkdevice.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Contact:<sip:applicationserver.broadsoft.com>
P-Asserted-Identity:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:none
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.15 Network to Cisco BroadWorks Network Server Call Redirection

The flow shown in *Figure 96* is the typical Cisco BroadWorks configuration. The network device is configured to send all messages to the Cisco BroadWorks Network Server. The Cisco BroadWorks Network Server then based on its provisioning, redirects the network device to the appropriate Cisco BroadWorks Application Server. The call setup between the network and Cisco BroadWorks Application Server is identical to the network to user flows previously described.

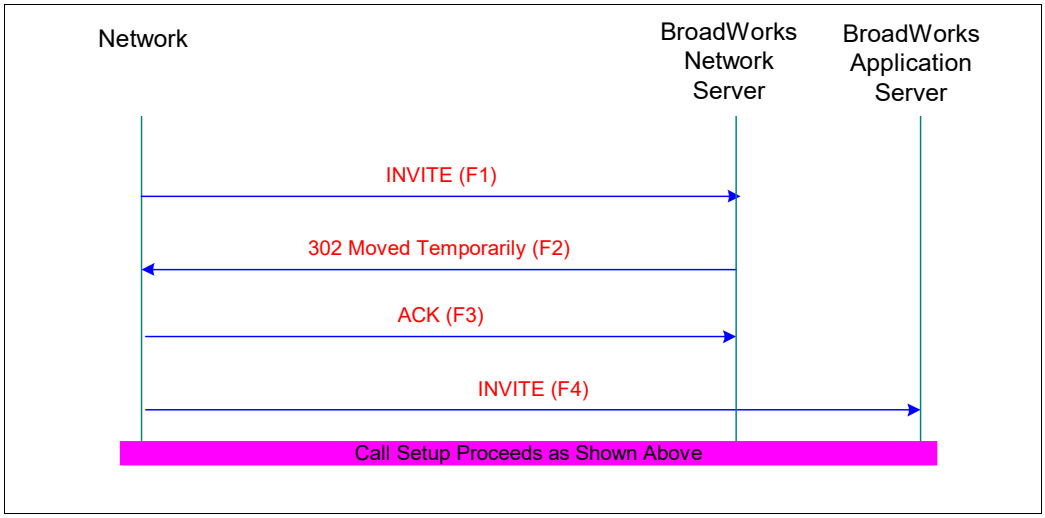

Figure 96 Network to Cisco BroadWorks Network Server Call Redirection

4.15.1 F1 – INVITE: Network → User

```
INVITE sip:+12403645111@networkserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkserver.broadsoft.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Contact: <sip:+13015400460@networkdevice.com;user=phone>
Max-Forwards:10
Supported:100rel,timer
Content-Type: application/sdp
Content-Length: 136

v=0
o=- 4851 3460 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20268 RTP/AVP 0
```

4.15.2 F2 – 302 Moved Temporarily: User → Network

```
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@ networkserver.broadsoft.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Content-Length:0
Contact: <sip:+12403645111@applicationserver.broadsoft.com;user=phone>
```

4.15.3 F3 – ACK: Network → User

```
ACK sip:networkserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith " <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkserver.broadsoft.com;user=phone>;tag=1954344904-985549204867
Call-ID: 1@networkdevice.com
CSeq: 101 ACK
Max-Forwards:10
Content-Length: 0
```

4.15.4 F4 – INVITE: Network → User

```
INVITE sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>;tag=2
To: <sip:+12403645111@networkserver.broadsoft.com;user=phone>
Call-ID: 2@networkdevice.com
CSeq: 102 INVITE
Contact: <sip:+13015400460@networkdevice.com;user=phone>
Max-Forwards:10
Supported:100rel,timer
Content-Type: application/sdp
Content-Length: 136

v=0
o=- 4851 3460 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20268 RTP/AVP 0
```

4.16 User to Network with Equal Access via Cisco BroadWorks Network Server

The flow shown in *Figure 97* is the typical Cisco BroadWorks configuration. Cisco BroadWorks is configured to send all messages for non-group calls to the Cisco BroadWorks Network Server. The Cisco BroadWorks Network Server then based on its provisioning, redirects the Cisco BroadWorks Application Server to the appropriate network device. The call setup between the Cisco BroadWorks Application Server and the network device is identical to the network to user flows previously described.

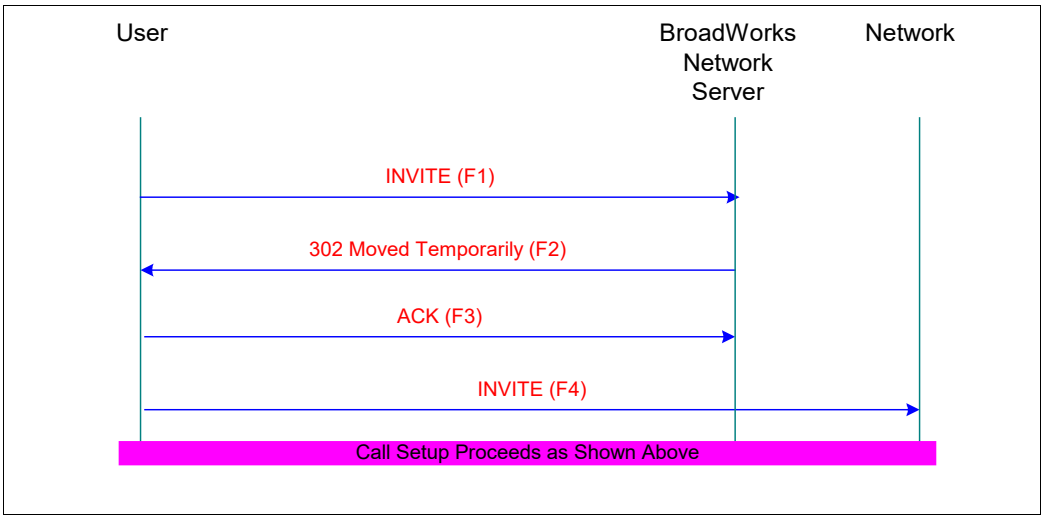

Figure 97 User to Network with Equal Access via Cisco BroadWorks Network Server

4.16.1 F1 – INVITE: User → Network

```
INVITE sip:19786641234@ networkserver.broadsoft.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
P-Asserted-Identity:”Joe Smith”<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:none
Contact:<sip:applicationserver.broadsoft.com>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.16.2 F2 – 302 Moved Temporarily: User → Network

```
SIP/2.0 302 Moved Temporarily
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Content-Length:0
Contact:<sip:19786641234;cic=+11234@networkdevice.com;user=phone>;q=0.5
```

4.16.3 F3 – ACK: Network → User

```
ACK sip:networkserver.broadsoft.com SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 ACK
Max-Forwards:10
Content-Length: 0
```

4.16.4 F4 – INVITE: User → Network

```
INVITE :sip:19786641234;cic=+11234@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1064300362 -
1815823761-990234856232
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1815823761-
990234856232
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1523581629385930101312432557900@applicationserver.broadworks.com
CSeq:1064300362 INVITE
```

```
Contact:<sip:applicationserver.broadsoft.com>
P-Asserted-Identity:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:none
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.17 User to Network with Originating Trunk Group via Cisco BroadWorks Network Server

The flow shown in *Figure 98* is the typical Cisco BroadWorks configuration. Cisco BroadWorks is configured to send all messages for non-group calls to the Cisco BroadWorks Network Server. The Cisco BroadWorks Network Server then based on its provisioning, redirects the Cisco BroadWorks Application Server to the appropriate network device. The call setup between the Cisco BroadWorks Application Server and the network device is identical to the network to user flows previously described. Note that the originating trunk group may also be included from the network into the user.

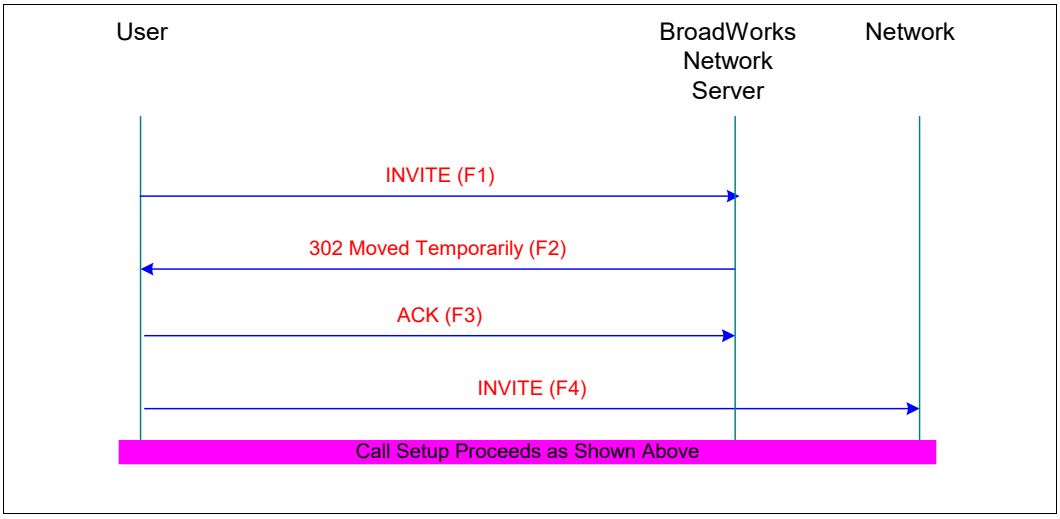

Figure 98 User to Network with Originating Trunk Group via Cisco BroadWorks Network Server

4.17.1 F1 – INVITE: User → Network

```
INVITE sip:19786641234@ networkserver.broadsoft.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
P-Asserted-Identity:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:none
Contact:<sip:applicationserver.broadsoft.com>
```

```
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.17.2 F2 – 302 Moved Temporarily: User → Network

```
SIP/2.0 302 Moved Temporarily
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 INVITE
Content-Length:0
Contact:<sip:19786641234@networkdevice.com;user=phone?P-Asserted-
Identity=%22Joe%20Smith%22%3csip:+12403645111%40applicationserver.broadsoft.com%3buser%3dphone%
3botg%3dotgsite1%3e>;q=0.5;ct=TO;ton=PUBLIC;cat=INTRALAT >;q=0.5
```

4.17.3 F3 – ACK: Network → User

```
ACK sip:networkserver.broadsoft.com SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
1646372935-985549141285
From:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>;tag=1646372935-
985549141285
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq:1000815249 ACK
Max-Forwards:10
Content-Length: 0
```

4.17.4 F4 – INVITE: User → Network

```
INVITE :sip:19786641234@networkdevice.com;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1064300362 -
1815823761-990234856232
From:"Joe
Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone;otg=otgsite1>;tag=1815823761-
990234856232
To: <sip:19786641234@networkserver.broadsoft.com;user=phone>
Call-ID:BW1523581629385930101312432557900@applicationserver.broadworks.com
CSeq:1064300362 INVITE
Contact:<sip:applicationserver.broadsoft.com>
P-Asserted-Identity:"Joe Smith"<sip:+12403645111@applicationserver.broadsoft.com;user=phone>
Privacy:none
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel, timer
Accept::application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:200
```

```
v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.18 Network to Cisco BroadWorks Subscription (Generic-Event Event Package Example)

The flow shown in *Figure 99* is the typical Cisco BroadWorks configuration. The network device is configured to send all messages to the Cisco BroadWorks Network Server. The Cisco BroadWorks Network Server then based on its provisioning, redirects the network device to the appropriate Cisco BroadWorks Application Server. The subscription is then processed by the Cisco BroadWorks Application Server and the initial state of the subscription is updated.

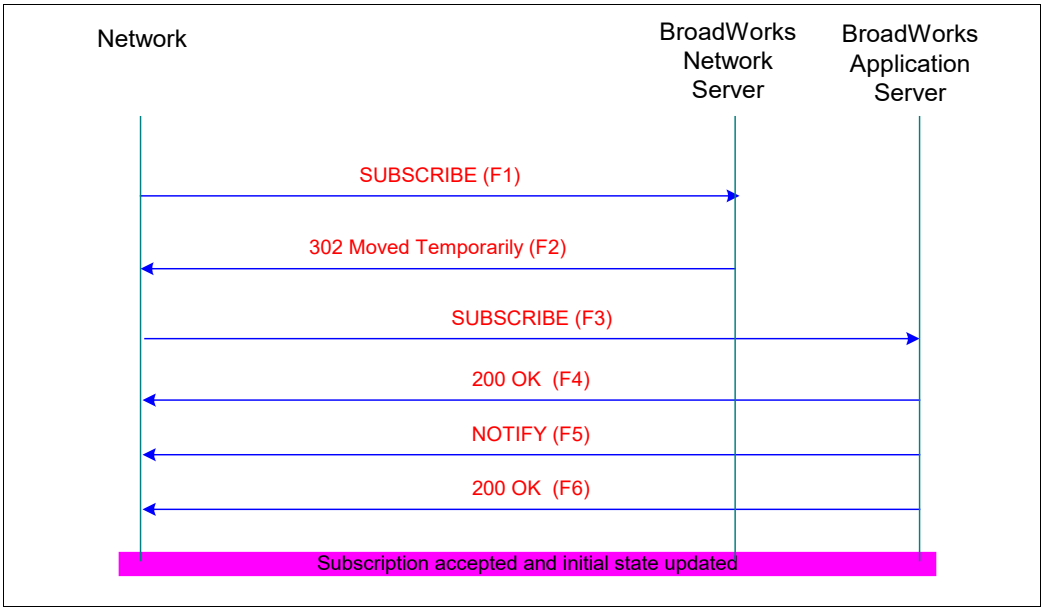

Figure 99 Network to Cisco BroadWorks Subscription (Generic-Event Event Package Example)

4.18.1 F1 – SUBSCRIBE: Network → User

```
SUBSCRIBE sip:group@broadsoft.com SIP/2.0
Via: SIP/2.0/UDP 10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=1
To: <sip:group@broadsoft.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 SUBSCRIBE
Contact: <sip:networkdevice@networkdevice.com;user=phone>
Max-Forwards:70
Expires: 3600
Event: generic-event
Accept: generic-event/info
Content-Length: 0
```

4.18.2 F2 – 302 Moved Temporarily: User → Network

```
SIP/2.0 302 Moved Temporarily
```

```
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=1
To: <sip:group@.broadsoft.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 SUBSCRIBE
Content-Length:0
Contact: <sip:group@broadsoft.com;maddr=applicationserver.broadsoft.com>q=0.5
```

4.18.3 F3 – SUBSCRIBE: Network → User

```
SUBSCRIBE sip:group@broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=2
To: <sip:group@broadsoft.com;user=phone>
Call-ID: 2@networkdevice.com
CSeq: 102 SUBSCRIBE
Contact: <sip:networkdevice@networkdevice.com >
Max-Forwards:70
Expires: 3600
Event: generic-event
Accept: generic-event/info
Content-Length: 0
```

4.18.4 F4 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=2
To: <sip:group@broadsoft.com;user=phone>;tag=13429028002-569234789
Call-ID: 2@networkdevice.com
CSeq: 102 SUBSCRIBE
Content-Length:0
Contact: <sip:applicationserver.broadsoft.com>
```

4.18.5 F5 – NOTIFY: User → Network

```
NOTIFY sip:networkdevice@networkdevice.com SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
13429028002-569234789
From: <sip:group@networkserver.broadsoft.com;user=phone>;tag=13429028002-569234789
To: "Network Device" <sip:networkdevice@networkdevice.com>;tag=2
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 NOTIFY
Contact: : <sip:applicationserver.broadsoft.com>
Max-Forwards:70
Expires: 3600
Event: generic-event
Subscription-State: active
Content-Type: generic-event/info
Content-Length: 111
Generic-event-info: example-body
```

4.18.6 F6 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-10.10.180.72V5060-0-1000815249 -
13429028002-569234789
From: <sip:group@networkserver.broadsoft.com;user=phone>;tag=13429028002-569234789
To: "Network Device" <sip:networkdevice@networkdevice.com>;tag=2
Call-ID:BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 1000815249 NOTIFY
Content-Length: 0
```

4.19 Cisco BroadWorks to Network Subscription (Calling-Name Event Package Example)

The flow shown in *Figure 100* occurs when the Cisco BroadWorks subscriber has the Calling Name service and the calling name was not available in the INVITE. The INVITE from the network device was redirected by the Network Server to the Application Server. This flow is shown as an example of a typical scenario where the Cisco BroadWorks Application Server subscribes to a network device.

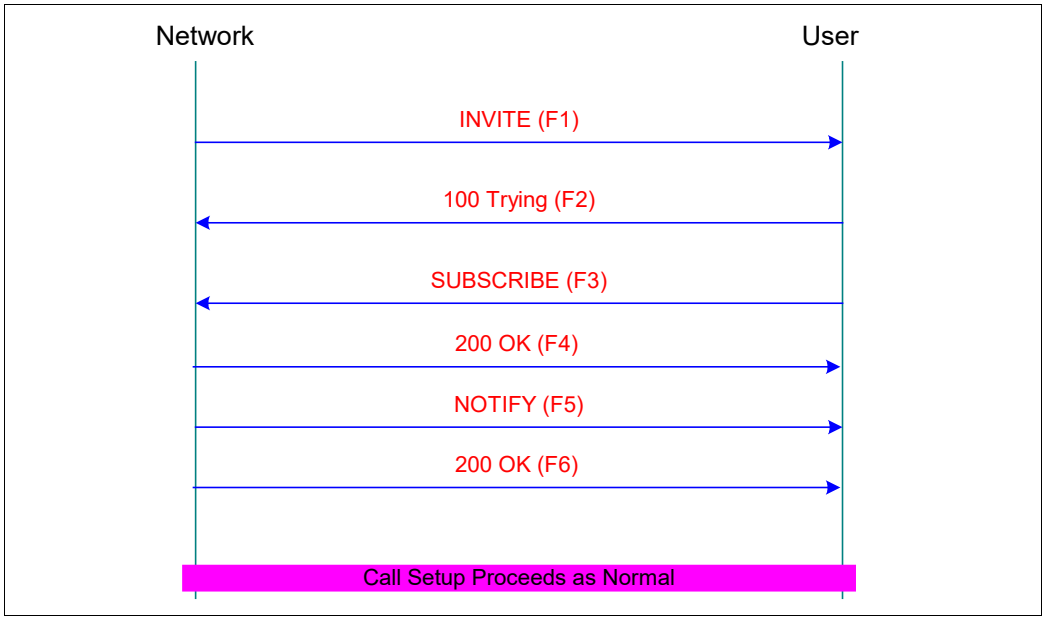

Figure 100 Cisco BroadWorks to Network Subscription (Calling-Name Event Package Example)

4.19.1 F1 – INVITE: Network → User

```
INVITE sip:+12403645111@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.10.180.72:54938
From: <sip:+13015400460@networkdevice.com;user=phone>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
Contact: <sip:+13015400460@networkdevice.com;user=phone>
Max-Forwards:10
Supported:100rel,timer
Content-Type: application/sdp
Content-Length: 136

v=0
o=- 4851 3460 IN IP4 10.10.180.72
s=SIP Call
c=IN IP4 10.10.180.72
t=0 0
m=audio 20268 RTP/AVP 0
```

4.19.2 F2 – 100 Trying: User → Network

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.180.72:54938
From: <sip:+13015400460@networkdevice.com>;tag=1
To: <sip:+12403645111@networkdevice.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 INVITE
```

```
Content-Length:0
```

4.19.3 F3 – SUBSCRIBE: User → Network

```
SUBSCRIBE sip:cnam.softswitch.com SIP/2.0
Via:SIP/2.0/UDP applicationserver.broadsoft.com:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.1.60V5060-0-703070627-536157426-1045083194181;event=calling-name
From: <sip:applicationserver.broadsoft.com>;tag=536157426-1045083194181
To: <sip:cnam.softswitch.com>
Call-Id: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 703070627 SUBSCRIBE
Contact: <sip:applicationserver.broadsoft.com>
Max-Forwards:70
Expires: 0
Event: calling-name;id=6
Content-Type: application/calling-name-info
Content-Length: 106

Calling-party:sip:+13015400460@networkdevice.com;user=phone
Called-party:sip:+12403645111@applicationserver.broadsoft.com;user=phone
```

4.19.4 F4 – 200 OK: Network → User

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP applicationserver.broadsoft.com:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.1.60V5060-0-703070627-536157426-1045083194181;event=calling-name
From: <sip:applicationserver.broadsoft.com>;tag=536157426-1045083194181
To: <sip:cnam.softswitch.com>;tag=4442
Call-Id: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 703070627 SUBSCRIBE
Expires: 0
Content-Length:0
Contact: <sip:cnam.softswitch.com>
```

4.19.5 F5 – Notify: Network → User

```
NOTIFY sip:applicationserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: <sip:cnam.softswitch.com>;tag=4442
To: <sip:applicationserver.broadsoft.com>;tag=536157426-1045083194181
Call-Id: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 20 NOTIFY
Contact: <sip:cnam.softswitch.com>
Max-Forwards:70
Event: calling-name;id=6
Subscription-State: terminated
Content-Type: application/calling-name-info
Content-Length: 131

Calling-Name-Status: available
Calling-Name: "Bob Smith" <sip:+13015400460@networkdevice.com;user=phone>
Presentation-Indicator: allowed
```

4.19.6 F6 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: <sip:cnam.softswitch.com>;tag=4442
To: <sip:applicationserver.broadsoft.com>;tag=536157426-1045083194181
Call-Id: BW1439010285250301013124125576944@applicationserver.broadworks.com
CSeq: 20 NOTIFY
Content-Length:0
```

4.20 Network to Cisco BroadWorks Message Waiting Indication

The flow shown in *Figure 101* is the typical flow for a Third-Party Voice Mail system to send message waiting indications to the Cisco BroadWorks Application Server. Note that the subscription for the message waiting indication is implicit and the interface integrity is maintained with the Network Traffic Security feature and the network interface access control list.

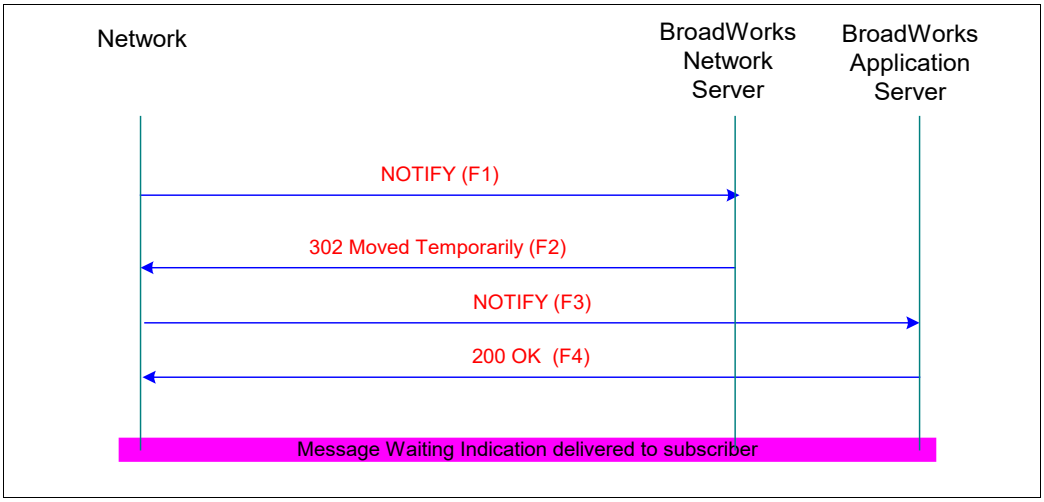

Figure 101 Network to Cisco BroadWorks Message Waiting Indication

4.20.1 F1 – Notify: Network → User

```
NOTIFY sip:3015400460@broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=1
To: <sip: 3015400460@broadsoft.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 NOTIFY
Contact: <sip:networkdevice@networkdevice.com;user=phone>
Max-Forwards:70
Event: message-summary
Subscription-State: terminated
Content-Type: application/simple-message-summary
Content-Length: 50
Messages-Waiting: yes
Voice-Message: 2/1 (0/0)
```

4.20.2 F2 – 302 Moved Temporarily: User → Network

```
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/UDP 10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=1
To: <sip: 3015400460@broadsoft.com;user=phone>
Call-ID: 1@networkdevice.com
CSeq: 101 NOTIFY
Content-Length:0
Contact: <sip:3015400460@applicationserver.broadsoft.com;user=phone>
```

4.20.3 F3 – NOTIFY: Network → User

```
NOTIFY sip:3015400460@applicationserver.broadsoft.com;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=2
To: < sip:3015400460@broadsoft.com;user=phone>
```

```
Call-ID: 2@networkdevice.com
CSeq: 102 NOTIFY
Contact: <sip:networkdevice@networkdevice.com>
Max-Forwards:70
Event: message-summary
Subscription-State: terminated
Content-Type: application/simple-message-summary
Content-Length: 50
Messages-Waiting: yes
Voice-Message: 2/1 (0/0)
```

4.20.4 F4 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Network Device" <sip:networkdevice@networkdevice.com>;tag=2
To: < sip:3015400460@broadsoft.com;user=phone>
Call-ID: 2@networkdevice.com
CSeq: 102 NOTIFY
Content-Length:0
```

4.21 Network to Cisco BroadWorks Instant Message

The flow shown in *Figure 102* is the typical flow for an instant message sent from a third-party proxy or Message server destined for Cisco BroadWorks. Note that the MESSAGE security on this interface is maintained with the Network Traffic Security feature and the network interface access control list.

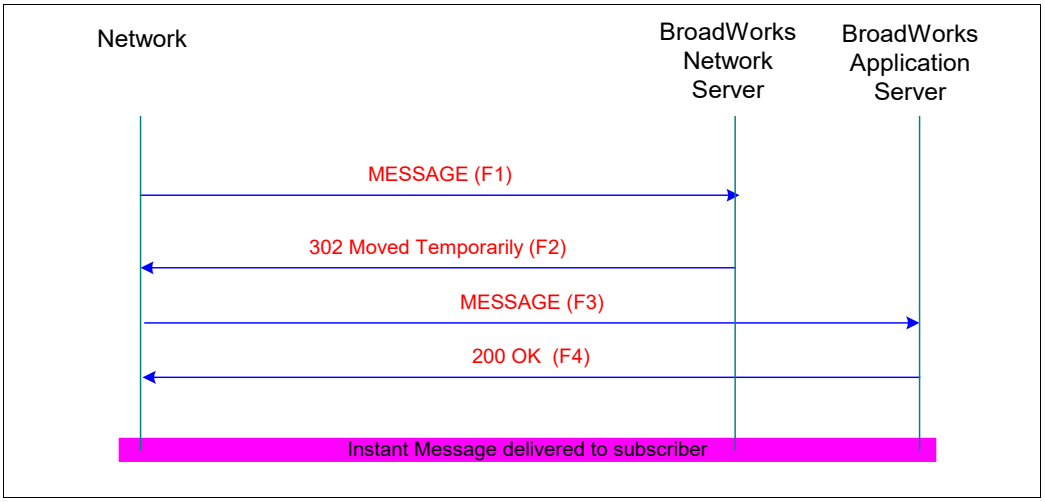

Figure 102 Network to Cisco BroadWorks Instant Message

4.21.1 F1 – MESSAGE: Network → User

```
MESSAGE sip:joe@broadsoft.com;user SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Alex Smith" <sip:alex@microsoft.com>;tag=1
To: <sip:joe@broadsoft.com>
Call-ID: 1@networkdevice.com
CSeq: 101 NOTIFY
Contact: <sip:networkdevice@networkdevice.com>
Max-Forwards:70
Content-Length: 8
Content-Type: text/plain
Content-Length: 18
Watson, come here.
```

4.21.2 F2 – 302 Moved Temporarily: User → Network

```
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Alex Smith" <sip:alex@microsoft.com>;tag=1
To: <sip:joe@broadsoft.com>
Call-ID: 1@networkdevice.com
CSeq: 101 NOTIFY
Content-Length:0
Contact: <sip:joe@broadsoft.com;maddr=applicationserver.broadsoft.com>q=0.5
```

4.21.3 F3 – MESSAGE: Network → User

```
MESSAGE sip:joe@broadsoft.com;maddr=applicationserver.broadsoft.com SIP/2.0
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Alex Smith" <sip:alex@microsoft.com>;tag=2
To: <sip:joe@broadsoft.com>
Call-ID: 2@networkdevice.com
CSeq: 102 NOTIFY
Contact: <sip:networkdevice@networkdevice.com>
Max-Forwards:70
Content-Length: 8
Content-Type: text/plain
Content-Length: 18
Watson, come here.
```

4.21.4 F4 – 200 OK: User → Network

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.10.180.72:54938
From: "Alex Smith" <sip:alex@microsoft.com>;tag=2
To: <sip:joe@broadsoft.com;user=phone>
Call-ID: 2@networkdevice.com
CSeq: 102 NOTIFY
Content-Length:0
```

5 Appendix A: SDP Overview

This section is a brief SDP overview. For a more detailed description of SDP, see *RFC 2327* http://www.ietf.org/rfc/rfc2327.txt. For a more detailed description of SDP use in SIP, see *RFC 3261* and *RFC 3264*.

SIP indicates the use of SDP by setting the entity-header Content-Type to "application/sdp", and entity-header Content-Length to the length of the SDP body. The SDP body starts at the end of the last *SIP* header and the last SIP header is followed by a blank line (nothing preceding a CRLF); the SDP body starts after this blank line.

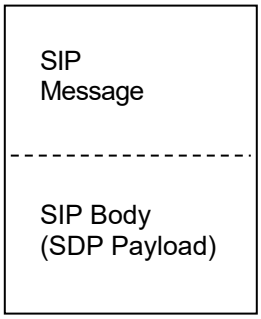

Figure 103 SDP Message

5.1 SDP Sections

There are three main sections in an SDP payload: Session Description, Timer Description, and Media Description.

All SDP sections consist of text lines in the following form, <type>=<value> pair, where <type> is always one lower case ASCII character. The <value> is a structured string whose format depends on the <type> field. No spaces are allowed on either side of the equals sign.

Some lines in each section are optional, but each line must appear according to the order shown in the following section.

5.1.1 Session Description

Session Description applies to the session being set up and all media streams being established. In SIP, only one Session Description is allowed per SIP message, although multiple SDPs can be sent per call leg. The last received SIP message with an SDP payload supersedes any earlier SDP messages.

Session Description Lines:

v= (Protocol version)

o= (Owner/Creator and session identifier)

s= (Session name)

i= (Session information) *Optional line*

u= (URL of description) *Optional line*

e= (e-mail address) *Optional line*

p= (Phone number) *Optional line*

c= (Connection information – Not needed if included in the media section)

b= (Bandwidth information) *Optional line*

** One or more Time Description lines (*see section *5.1.2 Timer Description)*

z= (Time zone adjustment) *Optional line*

k= (Encryption key) *Optional line*

a= (Session attribute lines) *Optional line*

** Zero or more Media Descriptions lines (*see section *5.1.3 Media Description)*

5.1.2 Timer Description

Timer Description defines the length of the session; it defines the start and stop times for the session being established.

Time Description Lines:

t= (Time the session is active) <start> <stop>

Start = 0, the session is permanent

Stop = 0, the session is not bound

r= (Repeat times) *Optional line*

5.1.3 Media Description

Media Description defines the type of media being transmitted (audio, video, data, and so on), the port listening on, protocol used, and the format of the media. The media section starts with the m= line, and contains other optional lines (as shown in the following list). If an optional line contains duplicate <types> from the Session Description, these duplicates override the Session Description value. In general, Session level values are the default, unless overridden by an equivalent media level value.

Multiple media sections are permitted; each new section starts with the next m= line.

Media Description Lines:

m= (Media name and transport address)

i= (Media title) *Optional line*

c= (Connection information is optional if in Session Description)

b= (Bandwidth information) *Optional line*

k= (Encryption key) *Optional line*

a= (Media attributes) *Optional line*

SIP SDP Example

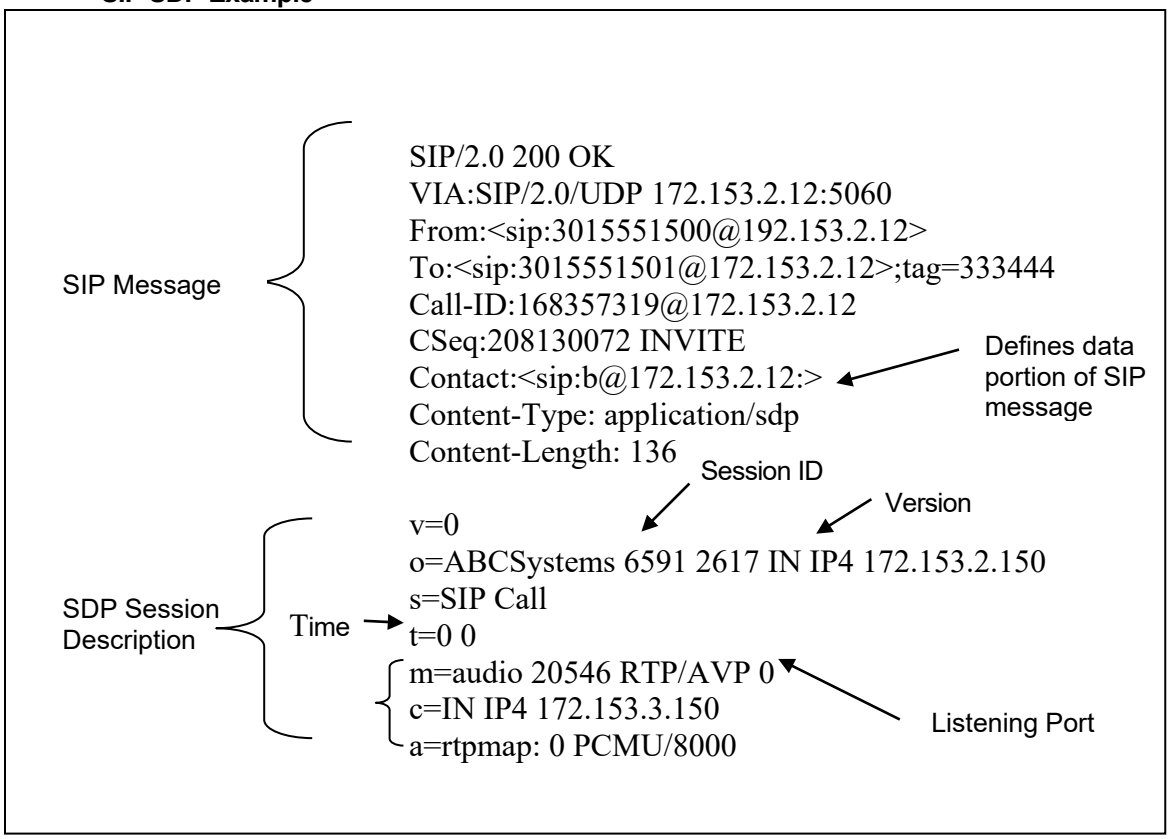

Figure 104 SDP Example

5.2 Caller to Callee SDP Media Setup

The caller and callee set up their media streams by aligning the media lines ("m=") in the Session Description. The nth m-line in the caller's SDP corresponds with the nth m-line in the callee's SDP. If the callee does not want to, or cannot support the media stream offered by the caller, the callee sets that port to zero, again keeping the "m" lines aligned.

Example:

Caller A to B

A → B

```
INVITE sip:B@192.168.2.12 SIP/2.0
VIA: SIP/2.0/UDP 192.168.2.12:5070
From: sip:<A@192.168.2.12>; tag=333999
To: <sip:B@192.168.2.12;user=phone>; tag=333444
Call-ID: 12345678@192.168.2.12:5070
Cseq:1 INVITE
Contact: <sip:192.168.2.12:5070;user=phone>
Content-Type: application/sdp
Content-Length: 124

v=0
o=A-System 12345 23456 IN IP4 172.174.34.85
s=Status Meeting
c=IN IP4 172.174.34.85
t=0 0
m=audio 20546 RTP/AVP 0
a=rtpmap: 0 PCMU/8000
a=sendrecv
m=video 20546 RTP/AVP 32
a=rtpmap: 32 MPV/90000
```

```
                    a=sendrecv
                    .
                    .  (Other SIP Messages)
                    .
B → A
                    SIP/2.0 200 OK
                    VIA:SIP/2.0/UDP 192.168.2.12:5060
                    From:<sip:A@192.168.2.12>; tag=333999
                    To:<sip:B@192.168.2.12>;tag=333444
                    Call-ID: 12345678@192.168.2.12:5070
                    CSeq: 1 INVITE
                    Contact:< sip: 192.168.2.12:>
                    Content-Type: APPLICATION/SDP
                    Content-Length: 136

                    v=0
                    o=B-Systems 6591 4897 IN IP4 172.153.2.150
                    s=SIP Call
                    t=0 0
                    c=IN IP4 172.153.2.150
                    m=audio 2087 RTP/AVP 0
                    a=rtpmap: 0 PCMU/8000
                    a=sendrecv
                    m=video 0 RTP/AVP 32
                    a=rtpmap: 32 MPV/90000
                    a=sendrecv
```

Figure 105 Example of Caller to Callee SDP Media Setup

The callee (B) rejects the video media stream by setting the port to zero.

5.3 Delayed Media Streams

If a caller does not know what media is supported at the time the call is initiated, it may delay its media lines (“m=”) in the SDP Session Description. By providing a Session Description, the callee knows that the caller wants to participate in a multimedia session, and would like the callee to supply supported media streams. The caller may update the Session Descriptions (media) either with an ACK or with a re-INVITE, at a later time.

To bring a call off hold, an SDP does not need to be included in the INVITE.

5.4 Adding and Deleting Media Streams

To add a media stream to an existing call, either party appends an additional “m” line to the previous Session Description when sending a re-INVITE. And to remove a media stream from a call, either side sends a re-INVITE and sets the port it wants to remove, to zero.

User Agents (UAs) should accept SDPs with “m” lines that are not aligned with earlier descriptions. If such a description is received, the “m” lines should be aligned based on the media types (audio, video). If a re-INVITE is received with “m” lines that do not synchronize, lines are omitted or added, and the UA may delete the added lines.

If a modification is made to the SDP Session Description (this includes the media line), the version field of the “o” line must be incremented. The version field is used to indicate that something has changed, and also to determine which SDP session is the most recent.

```
o=<user name> <session id> <version> <network type> <address type>
```

5.5 Putting Media Streams on Hold

To place a media stream on hold, send a re-INVITE with the same SDP Session Description as the original SDP, but add an attribute for the media stream of a=inactive and increment the version field of the “o” line.

6 Appendix B: Cisco BroadWorks Equal Access Support

"Equal Access" is based on the Modification of Final Judgment (MFJ) ruling by the United States Department of Justice concerning the divestiture of AT&T in 1982, which requires LECs to offer telecommunications carriers access to their networks "equal in type, quality and price to that supplied to AT&T and its subsidiaries". The FCC in its part 69 Access Rules mandated that equal access capability shall be available in all end office switches. In the Voice over IP paradigm, the softswitch platform is therefore also expected to support these access rules through IP-based interfaces. The main source of the equal access requirements is based on GR-690-Core.

Cisco BroadWorks provides support for equal access requirements through both the Application Server and the Network Server. The Application Server supports assigning of Preferred Inter-exchange Carriers (PICs) to individual users, groups, and/or enterprises (for enterprises, on a per-country code basis). The precedence for the call processing PIC selection for outgoing call setup messages is users, groups, or enterprises in that order. Thus, if a subscriber has a PIC assigned, then it overrides the PIC settings of the group and enterprise.

Figure 106 provides a general overview of carriers and shows how they are used in Cisco BroadWorks.

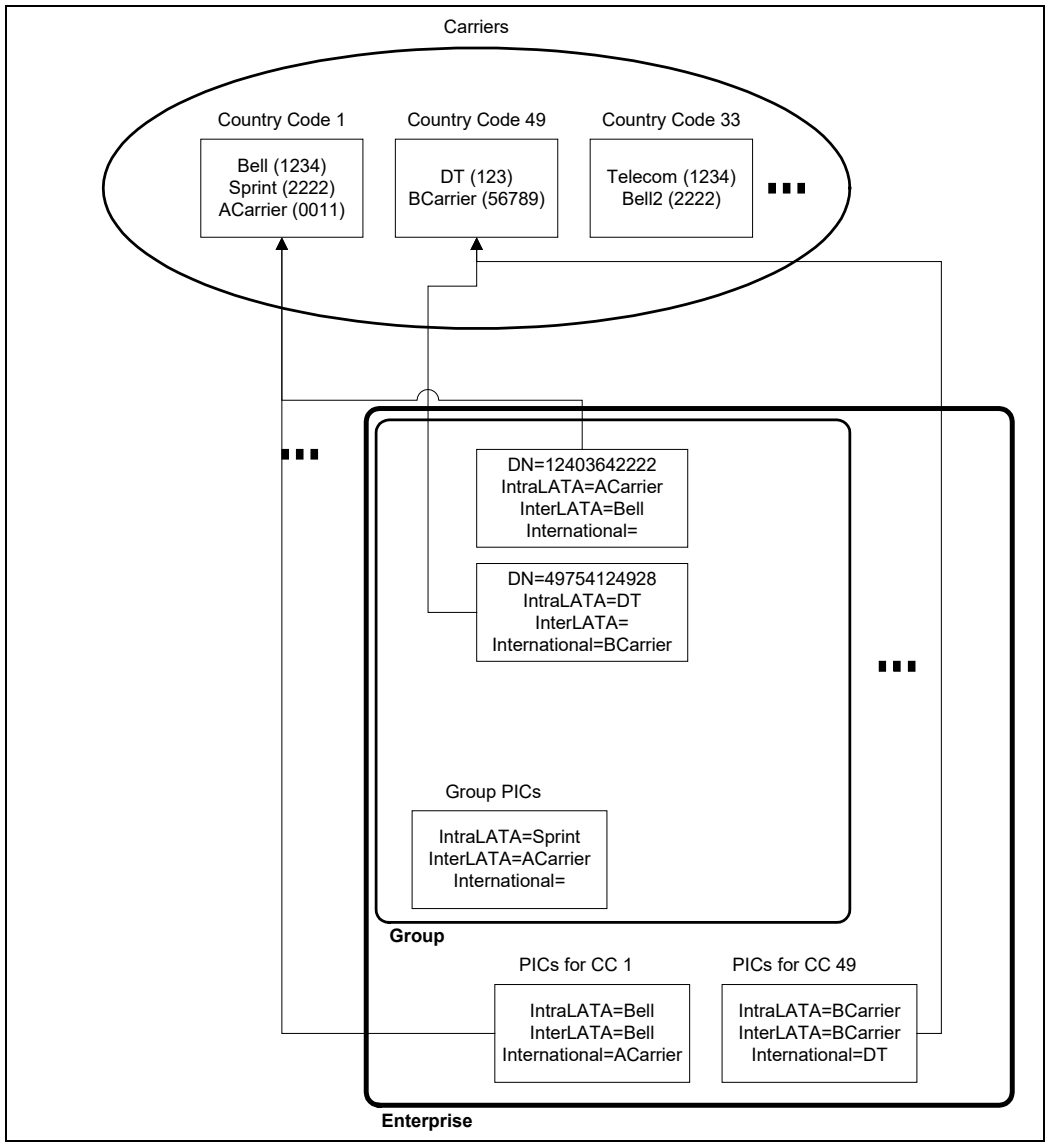

Figure 106 Overview of Carriers

6.1 Carrier

Cisco BroadWorks supports the concept of carrier(s) on a per-country code basis. A carrier entry is keyed using a system-wide unique name and it contains the following attributes:

- Country code that indicates the country in which this carrier exists
- Carrier Identification Code (CIC) (number up to 6-digits long) which is unique for each country code
- Indication (true/false) specifying if the carrier can be used to transport intraLATA calls
- Indication (true/false) specifying if the carrier can be used to transport interLATA calls

- Indication (true/false) specifying if the carrier can be used to transport international calls

Once a carrier is provisioned, it can be referenced by enterprises, groups, or users.

6.2 Enterprise

Cisco BroadWorks supports the creation of enterprise-PICs associations on a per-country code basis. This support includes the following type of associations, where “Z”, “Y”, and “Q” are carrier names in the system:

- Enterprise A uses carrier Z as its preferred carrier for callers in country code 1.
- Enterprise A has no preferred interLATA carrier defined for callers in country code 1.
- Enterprise A uses carrier Y as its preferred international carrier for callers in country code 1.
- Enterprise A has no preferred intraLATA carrier defined for callers in country code 49.
- Enterprise A uses carrier Q as its preferred interLATA carrier for callers in country code 49.
- Enterprise A uses carrier Q as its preferred international carrier for callers in country code 49.

An enterprise can have PICs defined for zero, one, or more country codes. However, only one set of PICs (intraLATA, interLATA, and international make a set) can be provisioned per-country code.

The same carrier can be used by more than one call category (intraLATA, interLATA, international), and by more than one enterprise at the same time.

For a given call category, a carrier can only be selected if this carrier is allowed to transport calls of this call category. For example, if “Sprint” is not an international carrier, it cannot be used as an international PIC in the system.

6.3 Group

Cisco BroadWorks supports the creation of an individual group-PICs association using the Preferred Carrier Group service. This service enables the following type of associations, where “Z” is a carrier name in the system:

- Group B uses carrier Z as its preferred intraLATA carrier for callers in its group.
- Group B has no preferred interLATA carrier defined for callers in its group.
- Group B uses the default preferred international carrier of its parent enterprise.

The same carrier can be used by more than one call category (intraLATA, interLATA, international) and by more than one group at the same time.

For a given call category, a carrier can only be selected if this carrier is allowed to transport calls of this call category. For example, if “Sprint” is not an international carrier, it cannot be used as an international PIC in the system.

When the Preferred Carrier Group service is assigned to a group, the following logic applies to select the initial setting for a carrier at provisioning time.

Condition	Result
No user in the group has the Preferred Carrier User service assigned.	The initial preferred carrier is None (feature disabled at group level).
At least one user in the group has the Preferred Carrier User service assigned, and at least one user refers to the “Use preferred group carrier” choice.	The initial preferred carrier is Use preferred service provider/enterprise carrier.
At least one user in the group has the Preferred Carrier User service assigned, but no user refers to the “Use preferred group carrier” choice.	The initial preferred carrier is None (feature disabled at group level).

6.4 User

Cisco BroadWorks supports the creation of an individual user-PICs association using the Preferred Carrier User service. This allows each individual user to have his or her own PICs. The feature enables the following type of associations, where “Z” is a carrier name in the system:

- User C has no preferred intraLATA carrier defined.
- User C uses carrier Z as User C’s preferred interLATA carrier.
- User C uses the default preferred international carrier of its parent group.

The same carrier can be used by more than one call category (intraLATA, interLATA, international) and by more than one user at the same time.

For a given call category, a carrier can only be selected if this carrier is allowed to transport calls of this call category. For example, if “Sprint” is not an international carrier, it cannot be used as an international PIC in the system.

When the Preferred Carrier User service is assigned to a user, the following logic applies to specify the initial setting for a carrier at provisioning time.

Condition	Result
Preferred Carrier Group service is not assigned.	The initial preferred carrier is None (feature disabled at user level).
Preferred Carrier Group service is assigned.	The initial preferred carrier is Use preferred group carrier.

6.5 Call Processing

Cisco BroadWorks supports the equal access requirements by including an indicator in the request-URI of outgoing SIP messages to the network, which contains the appropriate CIC for the particular call. The Application Server sends the carrier information in the *cic-related* parameter to Network Servers and other network elements in outgoing SIP INVITE messages.

Figure 107 provides an example equal access scenario:

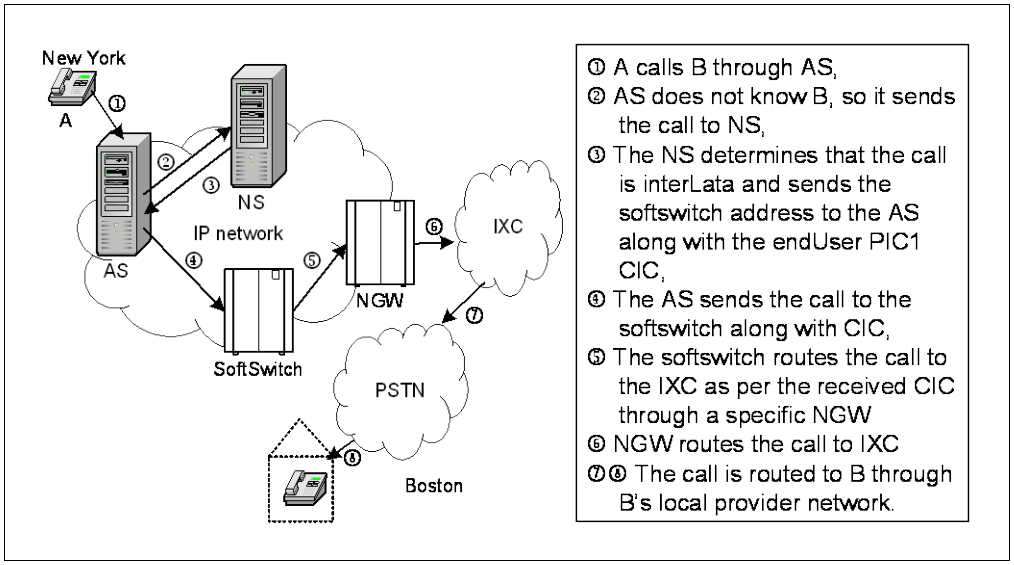

Figure 107 Sample Scenarios – A Calls B InterLATA

6.5.1 CIC-related Parameters and Processing

For outgoing calls to the network, the user part of the Request-URI, of the INVITE message contains the following *cic-related* parameters based on the conditions listed in the following table.

Condition	Result
▪ System-wide "csel" SIP option is enabled. ▪ Preferred Carrier User service and/or Group service are assigned. ▪ InterLATA carrier is found for the call, either in the Preferred Carrier User service data, in the Preferred Carrier Group service data, or in the default service provider settings (in that order).	cic1=+<country code of carrier><CIC of preferred interLATA carrier of caller>
▪ System-wide "csel" SIP option is enabled. ▪ Preferred Carrier User service and/or Group service are assigned. ▪ IntraLATA carrier is found for the call, either in the Preferred Carrier User service data, in the Preferred Carrier Group service data, or in the default service provider settings (in that order).	cic2=+<country code of carrier><CIC of preferred intraLATA carrier of caller>
▪ System-wide "csel" SIP option is enabled. ▪ Preferred Carrier User service and/or Group service are assigned. ▪ International carrier is found for the call, either in the Preferred Carrier User service data, in the Preferred Carrier Group service data, or in the default service provider settings (in that order).	cic3=+<country code of carrier><CIC of preferred international carrier of caller>

Condition	Result
▪ System-wide “csel” SIP option is enabled. ▪ Preferred Carrier User service and/or Group service are assigned. ▪ Application Server call type determination is successful (via Outgoing Calling Plan). ▪ A corresponding carrier is found for the call type, either in the Preferred Carrier User service data, in the Preferred Carrier Group service data, or in the default service provider settings (in that order).	cic=+<country code of carrier><CIC based on call type>

The “cic” represents the basic carrier selection result of the Application Server processing. The Network Server may honor it or replace it by another carrier, based on Network Server policies. The Application Server can only do very limited carrier selection without the Network Server. It can only choose between an international carrier or a toll carrier (always assumes interLATA toll).

If the PIC is set to “None” (that is, there is no preferred carrier defined) for a given call, the corresponding *cic[1|2|3]* parameters are omitted in the outgoing INVITE message. In addition, the *cic* parameter is not included if the “None” PIC found is for the call category of the call.

6.5.2 Carrier Selection (csel) Parameter and Processing

The *csel* parameter, *csel=<carrier selection indicator>*, is included in the outgoing INVITE message sent by the Application Server if the Preferred Carrier User service and/or Group service are assigned, and if the system-wide “csel” SIP option is enabled.

The *csel* parameter is defined by the following syntax.

```
csel = "noind" | "sub" | "subdialed" | "subnoind" | "dialed" | "ppref" |
"apref" | "verbalcgpty" | "verbalchgpty" | "emergency" | other-csel
other-csel=token
```

The Application Server generates the following values for the *csel* parameter:

- *sub*, if the CIC to be used was found at user, group, or enterprise level and no casual dialing was used
- *subdialed*, if the CIC to be used was found at user, group, or enterprise level and if the call originator dialed a carrier code that matches that CIC
- *dialed*, if the call originator dialed a carrier code that does not match the user, group, or enterprise CIC to be normally used for this call
- *noind*, in all other cases

6.5.3 Outgoing Calling Plan with Equal Access

Casual access calls can be blocked on the Application Server via the Outgoing Calling Plan (OCP) service. For all OCP call types for which the Application Server is unable to select a carrier (for example, local calls), the Application Server sends the *cic1, cic2*, and *cic3* parameters to the Network Server or network element, without any corresponding *cic* parameter. This allows the Network Server or network element to perform appropriate carrier selection even if a basic choice could not be performed by the Application Server itself.

The Application Server basic carrier selection capability is based on the Outgoing Calling Plan (OCP) call type of a call. It does not mean that the OCP feature needs to be assigned at user or group level for the carrier selection feature to be enabled. It simply means that system call types provisioned under *AS_CLI/Service/OutgoingCPCallTypes* are used as a means to find out if a call is Toll, International, Casual Call, or other.

6.5.4 Network Server Equal Access Processing

When a Cisco BroadWorks end user makes a phone call, the Application Server (AS) reports the dialed digits to the Network Server (NS) in an INVITE. The Network Server:

- Upon receipt of the INVITE, first identifies the call type of the call through its call typing policy.
- Using the call type, it then identifies the category of the call (using the Call Screening policy). The category can either be local, national, interLATA, intraLATA, international, or other. The Network Server uses a combination of the Telcordia LERG file (NNACL) and customer-defined local calling areas (LCAs) to determine the call category.
- When a call is identified as being either intraLATA, interLATA, or international, the Equal Access policy, determines which PIC to use along with a valid route.
- The information is then returned to the Application Server, which sends the call to a softswitch or gateway identified by the Network Server.

When a call is blocked by Network Server translations, a SIP error code is returned to the Application Server. The Application Server then provides the appropriate error treatment to the end user.

The Network Server may include the *cic* and *csel* parameters in the returned contact if the applicable policies for call translation are enabled on the Network Server. The *cic* and *csel* parameters are defined as follows:

- cic=+<country code of carrier><CIC based on call type>. The CIC is derived from the Network Server policies.
- csel=<carrier selection indicator>. The “csel” used is derived from the Network Server policies.

The Equal Access public routing policy on the Network Server processes incoming carrier information coming from the Application Server on a per-call basis. If the carrier is specified in the INVITE request sent to the Network Server, the Network Server uses the specified carrier. If the carrier to be used is unknown in the Network Server, the call continues with this carrier when the *BlockUnknownCac* parameter is “false”. Otherwise, the call is blocked. If the carrier to be used is known in the Network Server but is not allowed to transport a call of the current category, the call is blocked.

If the Application Server does not send carrier data for the call (for example, the call is interLATA but the Network Server did not receive cic1), existing Network Server processing continues using the policies for that group.

When the Network Server receives carrier information from the Application Server, the addCICasCAC option is ignored and addCICasCAC=false is assumed for these calls.

When the Network Server receives carrier information from the Application Server, the CicAlways option is ignored and CicAlways = true is assumed for these calls.

When the Network Server receives carrier information from the Application Server, the SIP interface parameter *GlobalCicEnabled* is ignored and *GlobalCicEnabled* = *true* is assumed for these calls.

The Network Server also uses the csel information to find out how the call was dialed. Equal Access uses the *csel* parameter to properly format the returned Contacts. The logic described in the following table is used is to populate the equal access information in the returned contacts.

Received csel	Received cic	Carrier Used by Equal Access	cic Returned	csel Returned
Sub	1234	1234	1234	Sub
Sub	1234	5555, found in cic1, cic2, or cic3	5555	Sub
Sub	1234	6666, not found in cic1, cic2, or cic3	6666	Noind
Subdialed	1234	1234, found in cic1, cic2, or cic3	1234	subdialed
Subdialed	1234	1234, not found in cic1, cic2, or cic3	1234	Dialed
Dialed	7777	7777, found in cic1, cic2, or cic3	7777	subdialed
Dialed	7777	7777, not found in cic1, cic2, or cic3	7777	Dialed
Noind	8888	Any, found in cic1, cic2, or cic3	Any	Subnoind
Noind	8888	Any, not found in cic1, cic2, or cic3	Any	Noind
-	-	Any, found in cic1, cic2, or cic3	Any	Sub

For all contacts generated by non-Equal Access policies, the original “csel” and “cic” values received from the Application Server are simply copied as is, in the user part of each contact.

The Equal Access public routing policy has the following capabilities:

- Support for casual access calls (CAC) (for example, 1010xxx calls):
  - CAC calls can also be denied on a per-carrier basis.
  - Network Server screens out unknown CICs.
- Support for Preferred Interexchange Carrier (PIC) calls:
  - PICs can be assigned to an enterprise or a specific group of users on the Network Server.
  - Support of intraLATA (PIC2), interLATA (PIC1), and international (PIC3) PICs.
- Support for operator-assisted calls: 0-, 00-, 0+, 00+
  - Calls are routed to PIC2 for 0- and 0+ calls, and PIC1 for 00- and 00+ calls.
- Support for system-level default PIC
- Support for cut-through calls (abbreviated dialing CAC+#):
  - Routing rules based on the dialed CAC.
- Support for special access code (SAC) calls:
  - By default, Cisco BroadWorks does not treat SAC (500, 8XX, 900) calls as equal access calls.
  - Special routes can be defined on the Network Server to handle those calls, that is, route them through a specific softswitch.
  - 700 calls are sent to PIC1.

The Network Server also may include the *lata telephone-subscriber* parameter in addition to the *cic* parameter, to identify the lata information for the call. The lata information is included by the Network Server under the following conditions:

- The CIC is known and valid.
- The LATA of the calling party is known and provisioned on the Network Server.

6.6 SIP-ISUP Mapping for GR-394 ISUP Parameters

Nature of Connection Indicators – No specific requirements from Cisco BroadWorks.

Forward Call Indicators – Bit D should be set to "1"; bit E, no requirements from Cisco BroadWorks; all others set to "0".

Calling Party's Category – Set to "ordinary calling subscriber" (00001010).

User Service Information – Set as per GR.317 for a non-ISDN subscriber (3.1 kHz Audio μ-law).

Called Party Number

- **Nature of Address Indicator** – Set based on the number in the user portion of the *Request-URI* header of the SIP INVITE from Cisco BroadWorks:
  - If the number is 7 digits, 1 + 7 digits, set to "subscriber number" (0000001).
  - If the number is 10 digits, 1+10 digits, matches the pattern N11 or 1N11 then set to "national number" (0000011). This is the default value.
  - If the number is an E.164 number (+ in front), then set to "international number" (0000100).
  - If number is 0 + 7 digits, then set to "subscriber number, operator requested" (1110001).
  - If the number is 0 + 10 digits, then set to "national number, operator requested" (1110010).
  - If the number starts with 01 or + 01 + CC + NDC, then set to "international number, operator requested" (1110011).
  - If the number is 0, 00, then set to "no number present, operator requested" (1110100).
- **Numbering Plan** – Set to "ISDN numbering plan" (001).
- **Address Information** – Set to the number in the user portion of the *Request-URI* header of the SIP INVITE from Cisco BroadWorks. The + is removed for E.164 numbers.

Calling Party Number

Nature of Address Indicator – Set based on the number in the user portion of the SIP URI, in the *P-Asserted-Identity* header of the SIP INVITE from Cisco BroadWorks.

  - If the country code of the number is the same as the serving country code, then set to "unique national number" (0000011).
  - If the country code is different, then set to "unique international number" (0000100).

- **Numbering Plan** – Set to "ISDN numbering plan" (001).
- **Presentation** – Set based on the presence of the *RFC 3323 Privacy* header with a value of ID, critical in the SIP INVITE from Cisco BroadWorks.

If not included or included with value “none”, then set to “presentation allowed” (00). Cisco BroadWorks only includes the values “id, critical” in the *Privacy* header when the calling party number presentation is restricted.

- **Screening** – Set to “network provided” (11).
- **Address Information** – Set to the number in the user portion of the SIP URI in the *P-Asserted-Identity* header in the SIP INVITE from Cisco BroadWorks. The + is removed for E.164 numbers.

NOTE: The *calling party number* parameter is not included if the *P-Asserted-Identity* header is not available, or the user portion is not included.

Charge Number

- **Nature of Address Indicator** – Set based on the number in the user portion of the SIP URI in the *P-Asserted-Identity* header in the SIP INVITE from Cisco BroadWorks.
  - Set to “ANI of the calling party; Subscriber number” (000 0001).
- **Numbering Plan** – Set to “ISDN numbering plan” (001).
- **Address Information** – Set to the number in the user portion of the SIP URI in the *P-Asserted-Identity* header in the SIP INVITE from Cisco BroadWorks. The + is removed for E.164 numbers.

Originating Line Information

- **Set to “Identified Line** – No special treatment”/ANI (0000 0000)”.

Transit Network Selection

- **Type of Network Identification** – Set to “national network identification” (010).
- **Network Identification Plan** – Set to “four-digit carrier identification code” (0010).
- **Digits** – Set to the *cic* parameter in the telephone-subscriber portion of the *Request-URI* header in the SIP INVITE from Cisco BroadWorks, if available.
- **Circuit Code** – Set to the circuit-code configured against the trunk group on the softswitch for non-international calls. For international signaling, the softswitch sets the circuit code to “international call, no operator requested” (0001), when 101XXXX(#) or 011 prefix is dialed, or sets the circuit code to “international call, operator requested” (0010), when 00, 101XXXX + 0(0), or 01 prefix is dialed.

Carrier Selection (assuming FG-D trunk)

- Set to “selected carrier identification code pre-subscribed and not input by calling party” (00000001).

Carrier Identification

- **Type of Network Identification** – Set to “national network identification” (010).
- **Network Identification Plan** – Set to “four-digit carrier identification code” (0010).
- **Digits** – Set to the *cic* parameter in the telephone-subscriber portion of the *Request-URI* header in the SIP INVITE from Cisco BroadWorks, if available.

7 Appendix C: SIP System Parameters

Name	Default Value	Allowed Values	Description	Reference
t1	500	Choice = {500, 1000, 2000, 5000, 7000, 9000}	This parameter specifies the SIP t1 timer (in milliseconds).	
t2	4000	Choice = {4000, 6000, 8000, 10000}	This parameter specifies the SIP t2 timer (in milliseconds).	
maxForwarding-Hops	10	Integer {0 to 2147483647}	This parameter specifies the maximum number of hops before a call is rejected because of too many forwards.	
invite-Authentication-Ratio	0.0	String {1 to 7 characters}	0, 1, 0.000000001 to 0.999999999; this is the ratio, expressed in decimals, of SIP INVITE messages that are to be challenged for authentication out of those sent to the AS from a Cisco BroadWorks user's originating device when this user has the Authentication service assigned.	SIP Early Media Management Enhancements, R14.0
encryptFrom-Header	false	Choice = {false, true}	When set to “true”, the *From* header is encrypted in the INVITEs sent to network devices when the caller requests privacy. When set to “false”, the *From* header is not encrypted in INVITEs sent to network devices irrespective of whether or not caller requests privacy.	
100rel	true	Choice = {false, true}	Allows delivery of provisional responses per *RFC 3262* to be activated or deactivated. “True” indicates that 100rel is supported by Cisco BroadWorks. “False” indicates 100rel is not supported by Cisco BroadWorks. Setting 100rel to “false” can improve capacity by preventing the PRACK message flow during call setup. However, this can degrade the user experience during UDP packet loss situations, when provisional responses such as *180 Ringing* are not successfully delivered. The default value is “true”.	CSCF Integration, R11.1 SIP Early Media Management Enhancements, R14.0

Name	Default Value	Allowed Values	Description	Reference
useDomainFor-Subscriber-Address	false	Choice = {false, true}	This parameter is used to control whether the Application Server uses the subscriber's domain for subscriber addressing. When set to “true”, Cisco BroadWorks uses the subscriber's domain for subscriber addressing (the host portion of the *From* header when the subscriber originates a call). In terminating scenarios, the host part of the FROM URI is also changed to the user domain if the remote address is a phone number. When set to “false”, Cisco BroadWorks uses the Application Server address for subscriber addressing. For example, when set to “true” and the originator's DN is +12403649004, the domain is broadworks.com, and the name is Bob Miller, then the *From* header for the originator is: From:"Bob Miller"<sip:+12403649004@broadworks.com;user=phone>;tag=1656954910-1080676512608. If set to “false” and the Application Server address is 192.168.0.53, then the *From* header for the originator is: From:"Bob Miller"<sip:+12403649004@192.168.0.53;user=phone>;tag=1656954910-1080676512608. The default value for this parameter is “false”. Note that this parameter should be set to “true” when cscfApplication is set to “true”.	CSCF Integration, R11.1 SIP Early Media Management Enhancements, R14.0
accessControl	false	Choice = {false, true}	When set to “false”, Cisco BroadWorks accepts an INVITE event if the domain name of the originator's location does not resolve to the same IP address of the UDP socket on which the message was received. When set to “true”, Cisco BroadWorks rejects an INVITE event if the domain name of the originator's location does not resolve to the same IP address of the UDP socket on which the message was received.	

Name	Default Value	Allowed Values	Description	Reference
sendE164	false	Choice = {false, true}	When set to “false”, Cisco BroadWorks uses all digits in the user portion of SIP URLs when sending INVITE messages for digit-dialed calls. When set to “true”, Cisco BroadWorks uses E.164 formatted strings digits in the user portion of SIP URLs when sending INVITE messages for digit-dialed calls.	
suspicious-Address-Threshold	3	Integer {0 to 11}	Defines the number of unacknowledged tries when sending to a given IP address associated with a fully qualified domain name (FQDN) before another IP address associated with the FQDN is tried. A value of “0” indicates that only one IP address from the DNS query shall be tried. During call setup when the routing tables are involved, */System/CallP/Routing/RouteParms/routeTimerLength* shall be used instead of suspicious AddressThreshold to advance among hostnames multiple IP addresses.	
privacyVersion	RFC3323	Choice = {RFC3323, RFC3323-Japan, privacy-03, privacy-00, none}	When set to “RFC3323”, Cisco BroadWorks supports the *P-Asserted-Identity* and *Privacy* headers described in *RFC 3325* and *RFC 3323*. When set to “RFC3323-Japan”, Cisco BroadWorks adds two *P-Asserted-Identity* headers to INVITEs transmitted to a trusted network. One *P-Asserted-Identity* header with a SIP-URL and the other with a TEL-URL. When set to “privacy-03”, Cisco BroadWorks supports the calling-line privacy using the methods described in *draft-ietf-sip-privacy-03*. This is the default value and should be used unless the network devices in your deployment do not support this. When set to “privacy-00”, Cisco BroadWorks supports calling line privacy using the methods described in *draft-ietf-sip-privacy-00*. When set to “none”, Cisco BroadWorks will not use any SIP headers that are used to require or support privacy. “encryptFrom Header” may be configured independently to block the calling line ID by setting its value to “true’.	

Name	Default Value	Allowed Values	Description	Reference
privacyEnforce-Screening	false	Choice = {false, true}	When set to “false”, indicates that the *Remote-Party-ID* header is to be used to determine the originating party information regardless of the value of the *screen* parameter. When set to “true”, if the message indicates that the originator's identity was not screened, then the originator is considered unscreened, PSTN, and identity unavailable.	
listeningPort	5060	Integer {1025 to 65355}	The port upon which the messages are received.	
networkProxy-Host		String {1 to 80 characters}	Indicates that network-bound SIP INVITE messages leaving Cisco BroadWorks should be sent to the specified location if its value is not an empty string.	
networkProxy-Port		Integer {1025 to 65355}	Indicates that network-bound SIP INVITE messages leaving Cisco BroadWorks should be sent to the specified location if its value is not an empty string.	
networkProxy-Transport	unspecified	Choice = {udp, tcp, unspecified}	Indicates the transport to use for sending to the specified location network-bound SIP INVITE messages leaving Cisco BroadWorks.	SIP TCP Enhancements, Rel 14.0
accessProxyHost		String {1 to 80 characters}	Defines the host/IP address. SIP Access Proxy is only used to indicate an alternate source from which it is acceptable to receive messages from an access device when ACL is enabled. Its use has no impact on how Cisco BroadWorks processes outbound messages.	
accessProxyPort		Integer {1025 to 65355}	Defines the port associated with the accessProxyHost. SIP Access Proxy is only used to indicate an alternate source from which it is acceptable to receive messages from an access device when ACL is enabled. Its use has no impact on how Cisco BroadWorks processes outbound messages.	
accessProxy-Transport	unspecified	Choice = {udp, tcp, unspecified}	Defines the transport associated with the accessProxyHost.	SIP TCP Enhancements, Rel 14.0

Name	Default Value	Allowed Values	Description	Reference
supportDnsSrv	true	Choice = {false, true}	When set to “false”, deactivates performing DNS SIP SRV queries. If Cisco BroadWorks is deployed in a network that has been provisioned without DNS SIP SRV records, setting the value to “false” avoids useless DNS SIP SRV queries. When set to “true”, activates performing DNS SIP SRV queries.	
maxAddresses-PerHostname	10	Integer {1 to 50}	Indicates the maximum number of addresses from a DNS query that shall be tried. The value includes both SRV queries and A record queries. Thus, if a SRV host name query results in multiple host names, the address count crosses multiple host names.	
maxAddresses-PerHostnameIn-Dialog	4	Integer {1 to 50}	Indicates the maximum number of addresses from a DNS query that shall be tried once a dialog has been established. The value includes both SRV queries and A record queries. Thus, if a SRV host name query results in multiple host names, the address count crosses multiple host names.	
useDomainFor-Realm	false	Choice = {false, true}	When set to “false”, the SIP authentication realm defaults to “Cisco BroadWorks” for all users. This is the default value. When set to “true”, the user's domain is used for the SIP authentication realm for all users.	
defaultRealm	BroadWorks			
includeT38-CapabilityInfo	false	Choice = {false, true}	When set to “false”, T38 capability information is not included for SIP devices. When set to “true”, T38 capability information is included for SIP devices.	
reInvite-Authentication	true	Choice = {false, true}	When set to “false”, indicates that Re-INVITES will not be authenticated. When set to “true”, SIP Re-INVITES will be authenticated.	SIP Early Media Management Enhancements, R14.0

Name	Default Value	Allowed Values	Description	Reference
supportAnswer-After	false	Choice = {false, true}	When set to “false”, an answer-after parameter is not sent. When set to “true”, CAP initiated SIP INVITES will be sent out with an answer-after=0 parameter.	
networkSupport-Gtd	false	Choice = {false, true}	When set to “false”, GTD signaling is not used on SIP messages. When set to “true”, indicates that the network supports GTD signaling on the appropriate SIP messages.	
privateDialPlan-OriginatorUses-Extension	false	Choice = {false, true}	When set to “false”, any SIP INVITES received with the private dialing plan context set (and it is not an intra-group call) will have the originating identification normalized to an E.164 number. Any *302* response received with the private dialing plan context set will cause the resulting SIP INVITE to be sent out with an extension phone number or E.164 number. When set to “true”, any SIP INVITE received with the private dialing plan context set will use an extension for the originator identification instead of normalizing the originator identification to an E.164 number if the originator identification is a phone number. Any *302* response received with the private dialing plan context set will cause the resulting SIP INVITE to be sent out with an extension instead of a phone number or E.164 number.	
disallowHolding-EmergencyCall	false	Choice = {false, true}	Indicates if users can place emergency calls on hold.	
enableHold-Normalization	true	Choice = {false, true}	Enable or disable *RFC 3264* hold Session Description Protocol (SDP) sending support. This serves to control how Cisco BroadWorks sends hold SDP on the network and access-side interfaces.	RFC 3264 Hold Enhancement, Rel 19.0

Name	Default Value	Allowed Values	Description	Reference
networkHold-Normalization	useRfc3264	Choice = {useUnspecifiedAddress, useInactive, useRfc3264}	Enable or disable RFC 3264 hold Session Description Protocol (SDP) sending support. This serves to control how Cisco BroadWorks sends hold SDP on the network and access-side interfaces.	
callingParty-CategoryFormat	none	Choice = {none, cpc, isup-oli, cpc-gtd}	This parameter selects the format used to send the calling party category parameter associated with a user. When set to “none”, the calling party category (*CPC*) parameter is not sent. When set to “cpc”, the *CPC* parameter is included in the *From* header, under the *CPC* field. When set to “isup-oli”, the *CPC* parameter is included in the *From* header, under the *isup-oli* field. When set to “cpc-gtd” the *CPC* parameter is included in the *GTD* header, under the *CPC* field.	Calling Party Category R11.0
networkSupport-InviteWithoutSdp	false	Choice = {false, true}	When this parameter is set to “true”, the network side supports INVITE requests without SDP and the Application Server allows INVITE requests to terminate to the network side without an SDP. When this parameter is set to “false”, the network side does not support INVITE requests without SDP and the Application Server includes the fake SDP in INVITE requests that do not have an SDP. In this case, the Application Server re-INVITEs the network endpoint immediately upon receiving the *200 OK* response to ensure that the network endpoint provides a valid offer SDP.	INVITE Without SDP, R11.1
symmetric-Signaling	false	Choice = {false, true}	When this parameter is set to “true”, Cisco BroadWorks is configured to send SIP messages on the same source tuple (protocol, IP address, port) as it uses to receive. Note that the system must be restarted for this change to be implemented.	Symmetric Signaling, R11.1
supportTcp	false	Choice = {false, true}	When this parameter is set to “true”, support to transport SIP signaling via TCP is enabled.	SIP TCP Support, R12.0
supportDnsNaptr	false	Choice = {false, true}	When this parameter is set to “true”, support for the use of DNS Naming Authority Pointer (NAPTR) records is enabled.	SIP TCP Support, R12.0

Name	Default Value	Allowed Values	Description	Reference
sendCarrier-Selection	false	Choice = {false, true}	When this parameter is set to “true”, the Application Server sends in outgoing Invite messages to the caller's preferred carrier identification codes (CICs) for intra-LATA (cic2), inter-LATA (cic1), and international calls (cic3). It also selects the carrier to be used (cic) based on the type of call.	Per-User Primary Inter/Intra-LATA Carriers, R12.0
sendDialedCAC	true	Choice = {false, true}	Controls whether the Application Server sends the dialed casual dial prefix and carrier code in the SIP INVITE requests to the Network Server.	Casual Call Dialing Enhancements, R17.0
originatingTrunk-GroupFormat	otg	Choice = {otg, x-nortel-profile}	When set to “otg”, the Application Server sends the originating trunk group in the *From* header as a SIP URI parameter. When set to “x-nortel-profile”, the Application Server sends the originating trunk group in the *X-Nortel-Profile* header.	CS2K Interop Support, R13.0
destinationTrunk-GroupFormat	dtg	Choice = {dtg, tgrpInContact, tgrpInRequestUri, x-nortel-profile}	When set to “dtg”, the Application Server sends the destination trunk group information in the outgoing INVITE as the *dtg* parameter of the *Request-URI*. When set to “tgrpInContact”, the destination trunk group information is passed in the outgoing INVITE as the *tgrp* and *trunk-context* parameters in the user portion of the *Contact* header URI. When set to “tgrpInRequestURI”, the destination trunk group information is passed in the outgoing INVITE as the *tgrp* and *trunk-context* parameters in the user portion of the *Request-URI*. When set to “x-nortel-profile”, an *X-Nortel-Profile* header is added to carry the destination trunk group information in the outgoing SIP INVITE sent towards the network.	Destination Trunk Group Support, Rel 14.sp3 Call Typing and RFC 4904 TGRP Support, R18.0
supportRFC3398	false	Choice = {false, true}	When this parameter is set to “false”, the Application Server allows remote to local ringback transitions. When this parameter is set to “true”, the Application Server does not allow remote to local ringback transitions unless it determines that the remote side is another Cisco BroadWorks server. The default value is “false”.	RFC 3398 Support, R13.0

Name	Default Value	Allowed Values	Description	Reference
restrictedDisplay-Name	Anonymous	String {1 to 80 characters}	Sets the system-wide SIP restricted display name. The default value is “Anonymous”.	Configurable Restricted Display Name, R13.0
maxNumberTcp-SocketsPer-System	1000	Integer {1 to 65355}	The maximum number of TCP sockets per system.	SIP TCP Enhancements, Rel 14.0
maxNumberTcp-SocketsPerPeer	100	Integer {1 to 65355}	The maximum number of TCP sockets per peer.	SIP TCP Enhancements, Rel 14.0
autoDiscard-StaleConnections	false	Choice = {false, true}	Specifies whether stale connections should be automatically discarded.	SIP TCP Enhancements, Rel 14.0
staleConnection-TimerInMinutes	60	Integer {1 to 15000}	Specifies how long a stale connection should be kept before being discarded.	SIP TCP Enhancements, Rel 14.0
treatDTMF-PoundAsFlash	true	Choice = {false, true}	Specifies whether DTMF pound should be treated as flash.	SIP Proxying Capability, R14.sp1
supportPEarly-MediaHeader	false	Choice = {false, true}	Specifies whether the Application Server proxies and generates the *P-Early-Media* header.	P-Early-Media Header, R14.sp1
sendDiversion-Inhibitor	false	Choice = {false, true}	Specifies whether a diversion inhibitor should be sent.	Diversion Inhibitor Enhancements, R14.sp1
networkSupport-Video	true	Choice = {false, true}	When this parameter is set to “true”, the network side supports video.	Network Side Video Offering Policy, R14.sp1
callingPartyE164-Normalization	system-CountryCode	Choice = {none, systemCountryCode, calledCountryCode}	This parameter configures how the Application Server normalizes the calling party's CLID (Calling Line Identity) for network originators. When set to “none”, the calling party's CLID for a network originator is never normalized to the E.164 format. When set to “systemCountryCode”, the calling party's CLID for a network originator is normalized to E.164 using the system country code. When set to “calledCountryCode”, the calling party's CLID for a network originator is normalized to E.164 using the called country code for the INVITE when the CLID is considered a phone number but is not already in E.164 format. The called country code is the country code for the called address.	IMS ISC Interoperability Enhancements, R14.sp1

Name	Default Value	Allowed Values	Description	Reference
supportRFC-3966Phone-Context	true	Choice = {false, true}	When the *supportRFC3966PhoneContext* parameter is set to “true”, the phone-context (if any) is used as specified in the IMS Interoperability Enhancements FS.	IMS ISC Interoperability Enhancements, R14.sp2
IncludePrivacy-User	true	Choice = {false, true}	This system parameter determines whether the Application Server includes the “user” value in the *Privacy* header when privacy is restricted.	IMS ISC Interoperability Enhancements, R14.sp2
broadworks-HoldingSDP-Method	holdSDP	Choice = {holdSDP, modifiedAddressSDP}	Specifies the holding Session Description Protocol (SDP) method system wide. The default value for a new installation is “holdSDP”. The default value for an upgrade is “modifiedAddressSDP”.	SIP Interface Enhancements, Rel 14.sp2
broadworks-HoldingSDPNet-Address		IP address (1 to 39 chars)	A dummy Ipv4 address to be used when building modified address SDPs.	SIP Interface Enhancements, Rel 14.sp2
broadworks-HoldingSDPIPv6-NetAddress		IP address (1 to 39 chars)	A dummy Ipv6 address to be used when building modified address SDPs.	Dual-Offer Support for Cisco BroadWorks, Rel 19.0
useStrictRFC-3264Compliance	false	Choice = {false, true}	When set to “true”, forces strict compliance to *RFC 3264* for SDP offer/answer.	SIP Interface Enhancements, Rel 14.sp2
disableSDP-ChangesFor-Answer-Responses	false	Choice = {false, true}	When set to “true”, disables SDP changes for answer responses.	SIP Interface Enhancements, Rel 14.sp2

Name	Default Value	Allowed Values	Description	Reference
accessForking-Support	multiple-Dialogs	Choice = {singleDialog, multipleDialogs}	This parameter specifies the forking support for the access side of the SIP interface. When set to “singleDialog”, the Application Server only supports single dialog on the access interface. When set to “multipleDialogs”, the Application Server supports multiple dialogs on the access interface.	Early Media UPDATE Method Support Functional Specification, Rel 18.0 Early Media and Preconditions Enhancements, Rel 23.0
accessSingle-DialogBehavior	singleDialog-WithUPDATE-IfAllowed	Choice = {singleDialog, singleDialogWithUPDATE, singleDialogWithUPDATEIfAllowed}	This parameter specifies single dialog behavior on the access interface. When set to “singleDialog”, all responses are mapped to the same dialog. In some cases, this means that the media session can be updated within the dialog on a subsequent 18x response or on the 2xx response. When set to “singleDialogWithUPDATE”, all responses are mapped to the same dialog. Also, media session updates on subsequent 18x responses are converted to the UPDATE method. When set to “singleDialogWithUPDATEIfAllowed”, all responses are mapped to the same dialog. Also, media session updates on subsequent 18x responses are converted to the UPDATE method when UPDATE is given in the Allow header; otherwise, the message is kept as an 18x response.	Early Media and Preconditions Enhancements, Rel 23.0

Name	Default Value	Allowed Values	Description	Reference
accessMultiple-DialogBehavior	multiple-DialogsWith-Error-Correction	Choice = {multipleDialogs, multipleDialogsWithErrorCorrection}	This parameter specifies the multiple dialog behavior for the access interface. When set to “multipleDialogsWithErrorCorrection”, the Application Server is generally transparent to the creation of SIP dialogs. In the case where the media session is updated within a given dialog on a subsequent 18x/2xx response, the Application Server forces the creation of a new SIP dialog. When set to “multipleDialogs”, the Application Server is transparent to the creation of SIP dialogs.	Early Media and Preconditions Enhancements, Rel 23.0
networkForking-Support	multiple-Dialogs	Choice = {singleDialog, multipleDialogs}	This parameter specifies forking support for network interface. When set to “singleDialog”, all responses are mapped to the same dialog. In some cases, this means that the media session may be updated within the dialog on a subsequent 18x response or on the 2xx response. When set to “multipleDialogs”, the Application Server is transparent to the creation of SIP dialogs.	Early Media UPDATE Method Support Functional Specification, Rel 18.0 Early Media and Preconditions Enhancements, Rel 23.0

Name	Default Value	Allowed Values	Description	Reference
networkSingle-DialogBehavior	singleDialog-WithUPDATE-IfAllowed	Choice = {singleDialog, singleDialogWithUPDATE, singleDialogWithUPDATEIfAllowed}	This parameter specifies single dialog behavior on the network interface. When set to “singleDialog”, all responses are mapped to the same dialog. In some cases, this means that the media session may be updated within the dialog on a subsequent 18x response or on the 2xx response. When set to “singleDialogWithUPDATE”, all responses are mapped to the same dialog. Also, media session updates on subsequent 18x responses are converted to the UPDATE method. When set to “singleDialogWithUPDATEIfAllowed”, all responses are mapped to the same dialog. Also, media session updates on subsequent 18x responses are converted to the UPDATE method when UPDATE is given in the *Allow* header; otherwise, the message is kept as an 18x response.	Early Media and Preconditions Enhancements, Rel 23.0
networkMultiple-DialogBehavior	multiple-DialogsWith-Error-Correction	Choice = {multipleDialogs, multipleDialogsWithErrorCorrection}	This parameter specifies the multiple dialogs behavior on the network interface. When set to “multipleDialogsWithErrorCorrection”, the Application Server is generally transparent to the creation of SIP dialogs. In the case where the media session is updated within a given dialog on a subsequent 18x/2xx response, the Application Server forces the creation of a new SIP dialog. When set to “multipleDialogs”, the Application Server is transparent to the creation of SIP dialogs.	Early Media and Preconditions Enhancements, Rel 23.0
proxyInfoInAllow-Header	false	Choice = {false, true}	Specifies whether *Allow* header proxying for the INFO method is enabled.	SIP Proxying Enhancements, Rel 14.sp6
proxyUpdateIn-AllowHeader	true	Choice = {false, true}	Specifies whether *Allow* header proxying for the UPDATE method is enabled.	SIP Proxying Enhancements, Rel 14.sp6

Name	Default Value	Allowed Values	Description	Reference
useSession-CompletionTimer	true	Choice = {false, true}	When set to “true”, the Application Server uses the SIP Session Completion timer set by the administrator via the CLI interface (sessionCompletionTimerMillis). Otherwise, the SIP Session Completion timer from t2 is used.	Configurable SIP Session Completion Timer, Rel 14.sp6
session-CompletionTimer	5000	Integer {5000 to 100000}	The value of the Session Completion timer in milliseconds.	Configurable SIP Session Completion Timer, Rel 14.sp6
useHistoryInfo-HeaderOn-NetworkSide	false	Choice = {false, true}	Specifies whether the *History Info* header should be used on the network side.	SIP Enhancements, Rel 15.0
requires-BroadWorksDigit-Collection	false	Choice = {false, true}	Determines whether Cisco BroadWorks is going to perform digit collection.	BroadWorks Network-side Digit Collection, R16.0
supportXFeature-Control	false	Choice = {false, true}	Determines whether the Application Server sends the *X-FeatureControl* header in SIP INVITE messages. When set to “true”, the *X-FeatureControl* header is inserted if the FAC programming result is known. If set to “false”, the *X-FeatureControl* header is never inserted.	X-FeatureControl for Call Forwarding Service, R16.0 Directed and All MSN Support for Call Forwarding, R17.0
chargeHeader-Format	charge-HeaderTel	Choice = {chargeHeaderSip, chargeHeaderTel, pChargeInfoSip, pChargeInfoTel, paiTelURI, diversionSip}	Determines whether the ChargeHeaderSip, ChargeHeaderTel, PChargeInfoSip, PChargeInfoTel, paiTelURI, or diversionSip header is used for the Charge Number service.	Configurable CLID in CDR Charging Enhancements, R16.0 Dialable Caller ID, R18.0 Asserted Identity Configuration Enhancements, R18.0 Charge Header Format Enhancements, Rel 19.0
noaValue	clgp-ani-natl-num	String {1 to 80 characters}	Indicates to the PSTN the nature of address.	Dialable Caller ID, R18.0
forceAnswer-SDPOnAnswer	true	Choice = {false, true}	Controls sending the SDP in the generated *200 OK* message when no SDP is received in the *200 OK* final response by the Application Server. The default value is “true”.	SIP Enhancements, R16.0
sendX-BroadWorks-DNCHeader	false	Choice = {false, true}	This system parameter controls whether or not the *X-BroadWorks-DNC* header is allowed to be included in SIP messages.	CLID and COLP Enhancements Functional Specification, Rel 16.0

Name	Default Value	Allowed Values	Description	Reference
encrypt-XBroadWorks-DNCHeader	false	Choice = {false, true}	This system parameter controls whether the *X-BroadWorks-DNC* header is encrypted when included in SIP messages.	CLID and COLP Enhancements Functional Specification, Rel 16.0
xBroadWorks-DNCHeaderKey	m!rUqR\x24T8Z7NgSyD	String {8 to 80 characters}	This system parameter contains the private encryption key that is used for encrypting the *X-BroadWorks-DNC* header in SIP messages.	CLID and COLP Enhancements Functional Specification, Rel 16.0
allow-BroadWorks-ConferenceInfo	false	Choice = {false, true}	This system parameter controls whether SIP INFO messages with an application/x-broadworks-conference-info+xml body are allowed.	CLID and COLP Enhancements Functional Specification, Rel 16.0
sendCallerName-InfoForNetwork-Calls	true	Choice = {false, true}	Controls whether Caller Name Information is sent in network calls.	Calling Name Retrieval Enhancements, Rel 17.0
include-Classmark	false	Choice = {false, true}	Controls the ability to proxy a classmark received over the SIP interface.	Classmark Information in SIP Messaging Functional Specification, Rel 17.0
send181-Response	false	Choice = {false, true}	Controls proxying and generating a SIP *181 Call is being forwarded* response.	Hidden Call Forwarding Option Support, Rel 17.0
routeToTrunking-DomainByDefault	false	Choice = {false, true}	This parameter is used to decide whether the trunking domain of a given trunk group should be used in outgoing requests.	Business Trunking Enhancements, Rel 17.0
clusterAddress		IP address | host | domain (1 to 80 chars)	The value for this parameter is used to populate the host part of Cisco BroadWorks user when identified by a phone number. If not set, and the publicIPAddress is set, publicIPAddress shall be used. Otherwise, the ipAddress from resolving "localhost" shall be used.	Session Data Replication, R17.0 Session Data Replication, R19.0
disabled-CLIDNumber-Value		String {1 to 80 characters}	Stores the user portion of the calling line identity when no calling line identity is enabled for a user.	IMS Calling Line Identity Delivery Enhancements, R18.0
suppressImplicit-Refer-Subscription	perRFC3515	Choice = {withoutNOTIFY, perRFC3515}	When set to "withoutNOTIFY", the Application Server suppresses the implicit REFER subscription without sending NOTIFY. When set to "perRFC3515", the Application Server sends NOTIFY per *RFC 3515* to suppress the implicit REFER subscription.	SIP Implicit REFER Subscription, Rel 18.0

Name	Default Value	Allowed Values	Description	Reference
networkSend-IdentityInUpdate-AndReInvite	false	Choice = {false, true}	When set to “true”, indicates that the Application Server updates the network subscriber with the updated identity information in an UPDATE/Re-INVITE message. When set to “false”, Application Server does not send the updated identity information. The default is “false”.	Connected Identity Update Functional Specification, Rel 18.0
networkReceive-IdentityInUpdate-AndReInvite	false	Choice = {false, true}	When set to “true”, indicates that the Application Server accepts the identity information sent from the network in an UPDATE/Re-INVITE message. When set to “false”, the Application Server ignores the identity information. The default is “false”.	Connected Identity Update Functional Specification, Rel 18.0
enableTS29163-Compliance	true	Choice = {false, true}	When set to “true”, support is provided for multiple Calling Line Identities to comply with the rules defined in 3GPP TS 29.163.	Calling Line Identity Compliance Enhancements Functional Specification, Rel 18.0
redirecting-AssertedIdentity-Policy	asserted-Identity	Choice = {assertedIdentity, redirectingIdentity, includeRedirectingIdentity WhenAssertedNotAvailable }	When set to “redirectingIdentity”, the Application Server uses the asserted identity for redirection. When set to “useAssertedIdentity”, the asserted identity received will be proxied. When set to “includeRedirectingIdentityWhenAssertedNotAvailable”, the asserted identity received will be proxied, and will include the redirecting party's asserted identity when the asserted identity is not present.	Calling Line Identity Compliance Enhancements Functional Specification, Rel 18.0
useAsserted-IdentityFor-PrivateCLID	false	Choice = {false, true}	When set to “true”, the Application Server uses the asserted number when privacy is requested. When set to “false”, the asserted number is not used.	IMS ISC Interoperability Enhancements, R14.sp2 Calling Line Identity Compliance Enhancements Functional Specification, Rel 18.0
transferNetwork-CauseID	false	Choice = {false, true}	This parameter enables the transfer of the Network Asserted Identity received from the network to the outbound device.	SIP and MGCP Network Cause ID for the Japanese Market, Rel 18.0
sipIpVersion	ipv4	Choice = {ipv4, ipv6, both}	This parameter specifies the IP version used for SIP.	Dual-Offer Support for BroadWorks, Rel 19.0
networkHold-Normalization	useRfc3264	Choice = {useUnspecifiedAddress, useInactive, useRfc3264}	Specifies how the Application Server performs SDP normalization on the network side when attempting to hold the device media.	RFC 3264 Hold Enhancement, Rel 19.0

Name	Default Value	Allowed Values	Description	Reference
supportPrivacy-None	false	Choice = {false, true}	This parameter specifies whether the Application Server respects the “none” value in the *Privacy* header for a user origination. When set to “true” and the user origination is received with the *Privacy* header set to “none”, the user's CLID Delivery Blocking service setting is ignored. However, if the user dials the CLID Delivery Blocking per Call Feature Access Code (FAC), the privacy restrictions are enabled since the user explicitly requested privacy in the origination. When set to “false” and the user origination is received with the *Privacy* header set to “none”, the user's CLID Delivery Blocking service setting is applied, privacy restrictions are enabled, and the user has the service enabled. The default value is “false”.	IMS ISC Interoperability Enhancements, R14.sp2 Support for SIPconnect 1.1, Rel 19.0
reportAltc-Supported	false	Choice = {false, true}	This system parameter allows you to specify whether the Application Server supports the SIP alternate connection header.	Dual-Offer Support for BroadWorks, Rel 19.0
suppress-Unreliable-AlertingForIVR	true	Choice = {false, true}	When set to “true”, the application server suppresses the *18x* message for IVR calls.	Suppress Unreliable 18x for IVR Scenarios, Rel 19.0
sendCall-CorrelationID-Access	false	Choice = {false, true}	When set to “true”, the Call Correlation Identifier is included in SIP messages to Access-side devices.	Call Correlation Identifier, Rel 20.0
sendCall-CorrelationID-Network	false	Choice = {false, true}	When set to “true”, the Call Correlation Identifier is included in SIP messages within the network.	Call Correlation Identifier, Rel 20.0
redirection-HeaderPriority	historyInfo	Choice = {historyInfo, diversion}	This parameter determines which redirection header is prioritized when both headers (*History-Info* header and *Diversion* header) are present.	Redirection Headers Priority Functional Specification, Rel 20.0
disableCOLPFor-RFC3323	false	Choice = {false, true}	When set to “true”, the *P-Asserted-Identity* header will not be part of the connected identity messages when the privacy version is set to “RFC3323”.	SIP Enhancements, Rel 21.0

Name	Default Value	Allowed Values	Description	Reference
viaBranchToken		String {1 to 7 characters}	This parameter defines a custom branch token to recognize call terminations across different clusters. Note that the system must be restarted for this change to be implemented.	SIP Enhancements, Rel 21.0
enforceDevice-FeatureSync-PolicyForSingle-UserMode	false	Choice = {false, true}	When set to “true”, the Device Feature Synchronization policy is considered for users in single user mode. When set to “false”, the policy is not enforced for call processing purposes.	SIP Enhancements, Rel 21.0
enableDelay-QuickReInvite	false	Choice = {false, true}	When set to “true”, the SIP Invite is resent with a delay.	SIP Enhancements, Rel 21.0
delayQuick-ReInvite-Milliseconds	1000	Integer {100 to 10000}	This parameter defines the delay to wait in milliseconds before resending a SIP Invite.	SIP Enhancements, Rel 21.0
treatUnknown-DisplayNameAs-Unavailable	true	Choice = {false, true}	This parameter determines whether to treat as Unavailable the presentation identity when the *From* header's display name of a SIP invite is set to “Unknown”.	Treatment and Unknown Calling Name Enhancement, Rel 21.0
enableRFC6044	false	Choice = {false, true}	This parameter enforces the mapping between the *History-Info* and *Diversion* redirection headers.	Support Mapping of Diversion Information Per RFC 6044-4458, Rel 21.0
supportCause-Parameter	false	Choice = {false, true}	This parameter specifies that the *cause* URI parameter is accepted in incoming INVITE and applied to *History-Info* entries in outgoing INVITE.	Support Mapping of Diversion Information Per RFC 6044-4458, Rel 21.0
includeHistory-InfoInResponse	false	Choice = {false, true}	This parameter specifies that the *History-Info* header is included in the SIP response to the originating endpoint.	Support Mapping of Diversion Information Per RFC 6044-4458, Rel 21.0
includeHeader-LevelPrivacyParameter	false	Choice = {false, true}	This parameter determines whether to include the header parameter in the Privacy header when the originating caller requests calling line Id blocking.	SimRing to CS, Rel 22.0

Name	Default Value	Allowed Values	Description	Reference
suppressRFC-3312-Preconditions	always	Choice = {always, never, suppressIfSingleDialog}	This parameter specifies whether the Application Server removes the RFC 3312 precondition attributes present in the SDP before proxying the SIP offer message to the far end. When set to “always”, the Application Server removes the precondition attributes. When set to “suppressIfSingleDialog”, the Application Server removes the precondition attributes if the originating interface is determined to be single dialog. When set to “never”, the Application Server does not remove the precondition attributes.	Support of Precondition Suppression, Rel 22.0
pCharging-Function-Addresses-Format	RFC7315	Choice = {RFC3455, RFC7315}	This parameter determines the format for SIP header *P-Charging-Function-Addresses* when built using configured data. When set to “RFC7315”, *P-Charging-Function-Addresses* is built using the *RFC 7315* syntax. When set to “RFC3455”, P-Charging-Function-Addresses is built using the *RFC 3455* syntax.	SIP/ISC Enhancements, Rel 22.0
includeDirectory-NumberInPAI	true	Choice = {false, true}	A flag used for enabling or disabling the Directory Number in a *P-Asserted-Identity* (PAI) header feature.	TS 29.163 Interoperability Enhancements, Rel 22.0
transmitIR94-DeviceVideo-Capability	true	Choice = {false, true}	This parameter controls whether the received video media feature tag is transmitted in outgoing SIP messages to complete video capability exchange as defined in IR.94.	IR.94 Video Call Interoperability, Rel 22.0
enforceRTCP-Connectivity	false	Choice = {false, true}	This parameter controls whether the Application Server always connects devices to ensure that RTCP keep-alive mechanism or inactivity timeout does not tear down the session.	Communication Hold Enhancement for RTCP, Rel 23.0
proxyForking-Provisional-Responses	False	Choice = {false, true}	This parameter determines whether provisional responses from secondary endpoints/locations should be proxied or not.	Early Media and Preconditions Enhancements, Rel 23.0
supportNoFork-Option	False	Choice = {false, true}	This parameter enables/disables the support for the *Request-Disposition* header with no-fork option.	Early Media and Preconditions Enhancements, Rel 23.0

Name	Default Value	Allowed Values	Description	Reference
defaultPEarly-MediaValue	None	Choice = {sendonly, inactive, none}	This parameter specifies the default value for the *P-Early-Media* header added into first 18x response with SDP for a dialog when *P-Early-Media* header is not present. When set to "sendonly" or "inactive", *P-Early-Media* header with the value is added to the first 18x response with SDP. When set to "none", no *P-Early-Media* header is added to the first 18x response with SDP.	Early Media and Preconditions Enhancements, Rel 23.0
support199	false	Choice = {false, true}	This parameter enables/disables support for 199 Dialog Terminated response defined in *RFC 6228*.	Early Media and Preconditions Enhancements, Rel 23.0
useDomainFor-Unavailable-Identity	false	Choice = {false, true}	This parameter determines how Cisco BroadWorks forms an unavailable identity. When set to "false", Cisco BroadWorks uses the SIP URI sip:unavailable@unknown.invalid to form an unavailable identity. When set to "true", Cisco BroadWorks uses the call processing domain to form an unavailable identity. It has no effect if the SIP parameter *enableTS29163Compliance* is set to "true".	Container Option Conversions, Rel 23.0.
enableHybrid-Mode	false	Choice = {false, true}	This parameter indicates the Application Server operates in a Hybrid Application Server execution call model to enable integration with an IMS mobile network. When enabled, this parameter allows the capability to configure access mobility locations with IMS Public User Identities while all other locations are configured with NGN line/port addresses.	Hybrid Application Server for Mobility Deployments, Rel 23.0
supportHeader-LevelPrivacy	False	Choice = {false, true}	This parameter controls whether the Cisco BroadWorks Application Server supports the SIP Privacy header with option value header. This parameter can be set to "true" or "false". When set to "true", the Cisco BroadWorks Application Server treats the received *Privacy* header option request as requesting anonymous presentation. When set to "false", the Cisco BroadWorks Application Server ignores the requested *Privacy* header option.	Support for Header Option in Privacy Header for GSMA_IR.92_OIR, Rel 23.0

Name	Default Value	Allowed Values	Description	Reference
supportPath-Header	false	Choice = {false, true}	This parameter controls whether to store the path data provided by a REGISTER request on the Application Server within the associated contact information, to build a *Route* header for the initial requests to the device with the path data and to build a *Route* header for the Call Recording Platform device, Resource NE, and Routing NE.	Path Header Support, Rel 23.0

References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J, Sparks R., Handley, M., and Schooler, E., "SIP: Session Initiation Protocol", RFC 3261, Internet Engineering Task Force, June 2002. Available from http://www.ietf.org/.

[2] Vaha-Sipila, A., "URLs for Telephone Calls", RFC 2806, Internet Engineering Task Force, April, 2000. Available from http://www.ietf.org/.

[3] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004. Available from http://www.ietf.org/.

[4] Yu, J, "NP Parameters for the "tel" URI", draft-ietf-iptel-tel-np-10, May 24, 2006.

[5] Mahy, R., "The Calling Party's Category tel URI Parameter", draft-mahy-iptel-cpc-00, June 22, 2003.

[6] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, Internet Engineering Task Force, November 2002. Available from http://www.ietf.org/.

[7] Jennings, C., Peterson, J., Watson, M., "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, Internet Engineering Task Force, November 2002. Available from http://www.ietf.org/.

[8] Marshall, W., Ramakrishnan, K., Miller, E., Russell, G., Beser, B., Mannette, M., Steinbrenner, K., Oran, D., Andreasen, F., Pickens, J., Lalwaney, P., Fellows, J., Evans, D., Kelly, K., Watson, M., "SIP Extensions for Caller Identity and Privacy", draft-ietf-sip-privacy-03, May 20, 2001.

[9] Marshall, W., Ramakrishnan, K., Miller, E., Russell, G., Oran, D., Andreasen, F., Mannette, M., Steinbrenner, K., Beser, B., Pickens, J., Lalwaney, P., Fellows, J., Evans, D., Kelly, K., "SIP Extensions for Caller Identity and Privacy", draft-ietf-sip-privacy-00 (superseded draft), November, 2000.

[10] Levy, S., Byerly, B., Yang, J. R., "Diversion Indication in SIP", draft-levy-sip-diversion-08, August 25, 2004.

[11] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, Internet Engineering Task Force, September 2002. Available from http://www.ietf.org/.

[12] Rosenberg, J., Schulzrinne, H., "Reliability of Provisional Responses in the Session Initiation Protocol (SIP)", RFC 3262, Internet Engineering Task Force, June 2002. Available from http://www.ietf.org/.

[13] Donovan, S., Rosenberg J., "Session Timers in the Session Initiation Protocol (SIP)", RFC 4082, Internet Engineering Task Force, April 2005. Available from http://www.ietf.org/.

[14] Rosenberg, J., Schulzrinne, H., "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, Internet Engineering Task Force, June 2002. Available from http://www.ietf.org/.

[15] Mahy, R., Gurbani, V., Tate, B., "Connection Reuse in the Session Initiation Protocol (SIP)", draft-ietf-sip-connect-reuse-06, August 21, 2006.

[16] Rosenberg, J., Peterson, J., Schulzrinne, H., Camarillo, G., "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", RFC 3725, April 2004.

[17] Vemuri, A., Peterson, J., "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures", RFC 3372, Internet Engineering Task Force, September 2002. Available from http://www.ietf.org/.

[18] Camarillo, G., Roach, A., Peterson, J., Ong, L., "Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping", RFC 3398, Internet Engineering Task Force, December 2002. Available from http://www.ietf.org/.

[19] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, Internet Engineering Task Force, June 2002. Available from http://www.ietf.org/.

[20] Donovan, S., "The SIP INFO Method", RFC 2976, Internet Engineering Task Force, October, 2000. Available from http://www.ietf.org/.

[21] Mahy, R., "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", RFC 3842, Internet Engineering Task Force, August, 2004. Available from http://www.ietf.org/.

[22] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., Gurle, D., "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, Internet Engineering Task Force, December 2000. Available from http://www.ietf.org/.

[23] Schulzrinne, H., Petrack, S., "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals", RFC 2833, Internet Engineering Task Force, May 2000. Available from http://www.ietf.org/.

[24] ITU-T T.38. *Procedures for real-time Group 3 facsimile communication over IP networks*. September 2005.

[25] Handley, M., Jacobson, V., "SDP: Session Description Protocol", RFC 2327, Internet Engineering Task Force, April 1998. Available from http://www.ietf.org/.

[26] Olson, S., Camarillo, G., Roach, A., "Support for IPv6 in Session Description Protocol (SDP)", RFC 3266, Internet Engineering Task Force, June 2002. Available from http://www.ietf.org/.

[27] Rosenberg, J., Schulzrinne, H., "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, Internet Engineering Task Force, June 2002. Available from http://www.ietf.org/.

[28] Camarillo, G., "The Early Session Disposition Type for the Session Initiation Protocol (SIP)", RFC 3959, Internet Engineering Task Force, December 2004. Available from http://www.ietf.org/.

[29] Camarillo, G., Schulzrinne, H., "Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)", RFC 3960, Internet Engineering Task Force, December 2004. Available from http://www.ietf.org/.

[30] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, Internet Engineering Task Force, January 1996. Available from http://www.ietf.org/.

[31] Schulzrinne, H, "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 1890, Internet Engineering Task Force, January 1996. Available from http://www.ietf.org/.

[32] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, Internet Engineering Task Force, July 2003. Available from http://www.ietf.org/.

[33] Schulzrinne, H, Casner, S., "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 3551, Internet Engineering Task Force, July 2003. Available from http://www.ietf.org/.

[34] Garcia-Martin, M., Henrikson, E., Mills, D., "Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the Third-Generation Partnership Project (3GPP)", RFC 3455, Internet Engineering Task Force, January 2003. Available from http://www.ietf.org/.

[35] Cisco Systems, Inc. 2019. *Cisco BroadWorks SIP Access Side Extensions Interface Specification, Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[36] Cisco Systems, Inc. 2019. *Cisco BroadWorks Calling Name Interface Specification, Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[37] "Telcordia Technology Generic Requirements GR-690-Core", Issue 2, With Revision 1, November 1996.

[38] Levin, O., Even, R., Hagendorf, P., "XML Schema for Media Control", RFC 5168, Internet Engineering Task Force, March 2008. Available from http://www.ietf.org/.

[39] Ott, J., Sullivan, G., Wenger, S., Even, R., "RTP Payload Format for the 1998 Version of the ITU-T Rec. H.263 Video (H.263+)", draft-ietf-avt-rfc2429-bis-04.txt, December 30, 2004.

[40] Cisco Systems, Inc. 2019. *Cisco BroadWorks AS Mode ISC Interface Specification, Release 23.0.* Available from Cisco at xchange.broadsoft.com.

[41] NENA. *National Emergency Number Association (NENA) Interim VoIP Architecture for Enhanced 9-1-1 Services (i2)* Issues 1 Draft, August 5, 2005. Available from NENA at http://www.nena.org.

[42] Cisco Systems, Inc. 2019. *Cisco BroadWorks Redundancy Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[43] Ejzak, R. "Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media", RFC 5009, Internet Engineering Task Force, September 2007. Available from http://www.ietf.org/.

[44] H., Schulzrinne, D. Oran and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol (SIP)", RFC 3326, Internet Engineering Task Force, December 2002. Available from http://www.ietf.org/.

[45] Cisco Systems, Inc. 2019. *Cisco BroadWorks Treatment Guide, Release 23.0.* Available from Cisco at xchange.broadsoft.com.

[46] S. Wenger, Hannuksela, M.M., Stockhammer, T., Westerlund, M., Singer, D., "RTP Payload Format for H.264 Video", RFC 3984, Internet Engineering Task Force, February 2005. Available from http://www.ietf.org/.

[47] Cisco Systems, Inc. 2008. *Zone Calling Restrictions Feature Description, Release 15.0.* Available from Cisco Systems at xchange.broadsoft.com.

[48] Poetzl, J., Huelsemann, M., Stupka, J.-M., "Extensions to the Session Initiation Protocol (SIP) for the support of the Call Completion Services for the European Telecommunications Standards Institute", draft-poetzl-sipping-call-completion-02. Available from http://tools.ietf.org/html/draft-poetzl-sipping-call-completion-02.

[49] Barnes, M., "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 4244, Internet Engineering Task Force, November 2005. Available from http://www.ietf.org/.

[50] York, D., Voxeo, Asveren, T., and Sonus, "P-Charge-Info – A Private Header (P-Header) Extension to the Session Initiation Protocol (SIP)", draft-york-sipping-p-charge-info-06, Internet Engineering Task Force, February 2009. Available from http://www.ietf.org/.

[51] 3rd Generation Partnership Project (3GPP). 2007. *3GPP TS 24.647 v8.0.0 Advice of Charge (AoC)*. Available from www.3gpp.org/.

[52] 3rd Generation Partnership Project (3GPP). 2007. *3GPP TS 29.658 v2.1.0 SIP Transfer of IP Multimedia Service Tariff Information*. Available from www.3gpp.org/.

[53] ACB PROTOCOL, TECHNICAL REQUIREMENTS, TTS, ACB Protocol Spec, 710 00158 AAAA-DSEd. 05 19/7/98.

[54] Levy, S., Mohali, M., "Diversion Indication in SIP", RFC 5806, Internet Engineering Task Force, March 2010. Available from http://www.ietf.org/.

[55] Portman, L., Lum, H., Johnston, A., Hutton, A., "The SIP-based Media Recording Protocol (SIPREC)", draft-portman-siprec-protocol-03, Internet Engineering Task Force, March 1, 2011. Available from http://www.ietf.org.

[56] Ravindranath, R., Ravindran, P., Kyzivat, P., "Session Initiation Protocol (SIP) Recording Metadata Format", draft-ram-siprec-metadata-format-01, Internet Engineering Task Force, March 8, 2011. Available from http://www.ietf.org.

[57] Levin, O., Camarillo, G., "The Session Description Protocol (SDP) Label Attribute", RFC 4574, Internet Engineering Task Force, August 2006. Available from http://www.ietf.org.

[58] Holmberg, C., Burger, E., Kaplan, H., "Session Initiation Protocol (SIP) INFO Method and Package Framework", RFC 6086, Internet Engineering Task Force, January 2011. Available from http://www.ietf.org.

[59] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., Gurle, D., "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, Internet Engineering Task Force, December 2002. Available from http://www.ietf.org/.

[60] Andreasen, F., McKibben, B., and Marchall, B., "Private Session Initiation Protocol (SIP) Proxy-to-Proxy Extensions for Supporting the PacketCable Distributed Call Signaling Architecture ", RFC 5503, Internet Engineering Task Force, March 2009. Available from http://www.ietf.org/.

[61] Schulzrinne, H., Polk, J., "Communications Resource Priority for the Session Initiation Protocol (SIP)", RFC 4412, Internet Engineering Task Force, February 2006. Available from http://www.ietf.org/.

[62] Gurbani, V. and Jennings, C.., "Representing Trunk Groups in tel/sip Uniform Resource Identifiers (URIs)", RFC 4904, Internet Engineering Task Force, June 2007. Available from http://www.ietf.org/.

[63] Handley, M., Jacobson, V., Jacobson, V., "SDP: Session Description Protocol", RFC 4566, Internet Engineering Task Force, July 2006. Available from http://www.ietf.org/.

[64] Kawamura, S., Kawashima, M., "A Recommendation for IPv6 Address Text Representation", RFC 5952, Internet Engineering Task Force, August 2010. Available from http://www.ietf.org/.

[65] Gurbani, V., Carpenter, B., Tate, B., "Essential Correction for IPv6 ABNF and URI Comparison in RFC 3261", RFC 5954, Internet Engineering Task Force, August 2010. Available from http://www.ietf.org/.

[66] Boucadair, M., Kaplan, H., Gilman, R., Veikkolainen, S., "The Session Description Protocol (SDP) Alternate Connectivity (ALTC) Attribute", RFC 6947, Internet Engineering Task Force, May 2013. Available from http://www.ietf.org/.

[67] Cisco Systems, Inc. 2019. *Cisco BroadWorks Call Recording Interface Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[68] Rosenberg, J., Schulzrinne, H., Kyzivat, P., "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, Internet Engineering Task Force, August 2004. Available from http://www.ietf.org.

[69] GSMA. 2015. IR.94 IMS Profile for Conversational Video Service, Version 10.0. Available from http://www.gsma.com/.

Acronyms and Abbreviations

ABNF	Augmented Backus-Naur Format
ACB	Automatic Callback
ACL	Access Control List
ACM	Audio Compression Manager
ALTC	Alternate Connectivity
ANI	Access-Network-Info
AoC	Advice of Charge
AS	Application Server
CAC	Casual Access Calls
CCBS	Completion of Communications to Busy Subscribers
CDR	Call Detail Record
CIC	Carrier Identification Code
CIF	Common Intermediate Format
CLI	Command Line Interface
CLID	Calling Line Identity
CNAM	Calling Name
COLP	Connected Line Identification Presentation
CPC	Calling Party Category
CPG	Call Progress Message
CSCF	Call Session Control Function
DGC	Distributed Group Call
DN	Directory Number
DNC	Distributed Network Call
DNS	Domain Name System
DTG	Destination Trunk Group
DTMF	Dual-Tone Multi-Frequency
ESGW	Emergency Service Gateway
ESQK	Emergency Services Query Key
ESRN	Emergency Services Routing Number
FAC	Feature Access Code
FCC	Federal Communications Commission
FLP	Fixed Line Persona
FQDN	Fully Qualified Domain Name
FXS	Foreign eXchange Subscriber

GTD	Generic Transparency Descriptor
IAM	Initial Address Message
ICID	IMS Charging Identity
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISUP	Integrated Services User Part
IVR	Interactive Voice Response
LATA	Local Access Transport Area
LCA	Local Calling Area
LEC	Local Exchange Carrier
LERG	Local Exchange Routing Guide
MFJ	Modification of Final Judgment
MGCP	Media Gateway Control Protocol
MLP	Mobile Line Persona
MSCML	Media Server Control Markup Language
MSN	Multiple Subscriber Number
MWI	Message Waiting Indication
NAPTR	Naming Authority Pointer
NE	Network Element
NENA	National Emergency Number Association
NGN	Next Generation Network
NNACL	NPA-NXX Active Code List
NS	Network Server
OCP	Outgoing Calling Plan
OTG	Originating Trunk Group
PAI	P-Asserted-Identity
PBX	Private Branch Exchange
PIC	Preferred Inter-exchange Carrier
PPI	P-Preferred-Identity
PSAP	Public Safety Answering Point
PSTN	Public Switched Telephone Network
QCIF	Quarter Common Intermediate Format
RFC	Request for Comments

RPID	Remote-Party-ID
RTP	Real-Time Transport Protocol
SAC	Special Access Code
SBC	Session Border Controller
SDP	Session Description Protocol
SHLR	Smart Home Location Register
SIP	Session Initiation Protocol
SIPREC	SIP Recording
SMS	Short Message Service
SMSC	Short Message Service Center
SR	Selective Router
SRV	Service Locator
TCAP	Transactional Capabilities Application Part
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TLS	Transport Layer Security
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UDPTL	User Datagram Protocol Transport Layer
URI	Uniform Resource Identifier
VoIP	Voice over Internet Protocol

Index