Cisco BroadWorks

SIP Access 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

Copyright© 2019 Cisco Systems, Inc. All rights reserved.

Trademarks

Any product names mentioned in this document may be trademarks or registered trademarks of Cisco or their respective companies and are hereby acknowledged.

Document Revision History

Release	Version	Reason for Change	Date	Author
8.0	1.1	Introduced Release 8 changes.	July 19, 2002	Sam Hoffpauir
8.0	1.2	Changed location of revision number and matched to Razor.	July 19, 2002	Nora Navin
8.0	1.3	Made minor changes to references.	July 22, 2002	Sam Hoffpauir
8.0	1.4	Added section for Interface changes between Releases 7 and 8.	August 19, 2002	Sam Hoffpauir
9.0	2.1	Updated with Release 9 interface changes.	April 8, 2003	Sam Hoffpauir
9.0	2.2	Updated with *RFC 3261* loose routing support.	June 25, 2003	Sam Hoffpauir
9.0	2.3	Updated with Release 9.1 interface changes.	August 22, 2003	Sam Hoffpauir
10.0	1.0	Updated with Release 10 interface changes.	December 3, 2003	Sam Hoffpauir
11.0	1.0	Updated with Release 11 interface changes.	April 9, 2004	Sam Hoffpauir
12.0	1.0	Updated with Release 12 interface changes.	October 13, 2004	Sam Hoffpauir
12.0	2.0	Made minor editorial changes.	November 29, 2004	Patricia Renaud
12.0	3.0	Updated tel URI reference.	December 7, 2004	Sam Hoffpauir
12.0	4.0	Updated with *RFC 3263* clarification.	December 9, 2004	Sam Hoffpauir
12.0	5.0	Added video codec requirements.	January 10, 2004	Sam Hoffpauir
12.0	6.0	Added clarifications and made editorial fixes.	April 13, 2004	Sam Hoffpauir
13.0	1.0	Updated with Release 13 interface changes.	September 9, 2005	Sam Hoffpauir
14.0	1.0	Updated document for re-branding.	April 14, 2006	Roberta Boyle
14.0	1.0	Corrected support for authentication of SIP Notify method in section *3.33 BroadWorks Access Device Configuration Requirements*.	June 21, 2006	Robb Surridge
14.0	1.1	Updated with Release 14 Interface changes.	June 30, 2006	Sam Hoffpauir
14.0	1.1	Edited document.	September 26, 2006	Roberta Boyle
14.0	2.0	Updated document with fax information.	October 25, 2006	Stephane Bastien
14.0	2.0	Edited changes.	October 26, 2006	Patricia Renaud
14.0	3	Corrected section *3.8 Offer/Answer and Early Media Support*.	December 12, 2006	Martin Perron
14.0	3	Edited changes.	January 2, 2007	Patricia Renaud

©2019 CISCO SYSTEMS, INC.

Release	Version	Reason for Change	Date	Author
14.0	3	Added section *3.35 Deny Calls From Unregistered Users* for EV 41340.	February 3, 2007	Roberta Boyle
14.0	3	Edited changes.	February 16, 2007	Patricia Renaud
14.0	4	Updated section *3.18 Session Initiation Protocol (SIP) Refer Method (RFC 3515)/SIP "Replaces" Header (RFC 3891)/SIP Referred-BY Mechanism (RFC 3892)* for EV 41458.	March 22, 2007	Roberta Boyle
14.0	4	Edited changes.	March 30, 2007	Patricia Renaud
14.0	5	Updated section *3.2 SIP Subscriber Identification/Addressing* to clarify decision process (EV 46583).	May 15, 2007	Robb Surridge
14.0	5	Edited changes and published document.	June 12, 2007	Andrea Fitzwilliam
14.0	6	Updated document with Release 14.sp1 and Release 14.sp2 content.	August 27, 2007	Martin Piotte
14.0	6	Edited changes and published document.	October 1, 2007	Andrea Fitzwilliam
14.0	7	Updated document with Release 14.sp3 and Release 14.sp4 content.	December 20, 2007	Martin Piotte
14.0	7	Repaired lost figures and removed incorrect information.	May 8, 2008	Martin Piotte
14.0	7	Edited changes and published document.	May 13, 2008	Andrea Fitzwilliam
14.0	8	Updated document with Release 14.sp5 and Release 14.sp6 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 from Release 14 and published document for Release 15.0.	July 29, 2008	Andrea Fitzwilliam
15.0	2	Added description of the *Via* header and AA loop detection.	January 21, 2009	Martin Piotte
15.0	2	Added Enterprise Trunking enhancements (Release 15.sp2).	May 11, 2009	Martin Piotte
16.0	1	Updated document for Release 16.0.	May 19, 2009	Martin Piotte
16.0	1	Updated list of BroadWorks proprietary parameters for the *Diversion* and *History-Info* headers.	June 22, 2009	Martin Piotte
16.0	1	Edited changes for EV 95196 only.	June 26, 2009	Andrea Fitzwilliam
16.0	1	Edited changes and published document.	July 13, 2009	Andrea Fitzwilliam
16.0	1	Updated section *3.2 SIP Subscriber Identification/Addressing* for EV 95023.	August 13, 2009	Roberta Boyle
16.0	2	Updated provisional response section to add deviations for EV 104113 and EV 104164.	February 16, 2010	Eric Bernier
16.0	2	Edited changes and published document.	February 22, 2010	Andrea Fitzwilliam

©2019 CISCO SYSTEMS, INC.

Release	Version	Reason for Change	Date	Author
17.0	1	Updated document for Release 17.0.	April 8, 2010	Martin Piotte
17.0	1	Made minor corrections to diversion reasons and trunk identities.	April 20, 2010	Martin Piotte
17.0	1	Edited changes and published document.	April 20, 2010	Margot Hovey-Ritter
17.0	2	Made minor editorial corrections.	April 30, 2010	Patricia Renaud
17.0	2	Added that BroadWorks now adds the *Reason* header to the Ring Splash CANCEL message in section *3.7.1 Priority Ringing on Device and Ring Splash*) for EV 113116.	June 10, 2010	Martin Piotte
17.0	2	Edited changes and published document.	June 25, 2010	Andrea Fitzwilliam
17.0	3	Added information on SIP timers.	October 15, 2010	Martin Piotte
17.0	3	Edited changes and published document.	October 20, 2010	Andrea Fitzwilliam
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 22, 2012	Patricia Renaud
19.0	2	Corrected outdated information concerning the *maxHops* CLI parameter in section *3.6 Diversion Indication in SIP (RFC 5806)* for EV 177987.	November 5, 2012	Doug Sauder
19.0	2	Edited changes and published document.	December 18, 2012	Andrea Fitzwilliam
19.0	3	Removed appendix entitled *SIP Protocol Requirements for BroadWorks Features* for EV 177212.	March 27, 2013	Doug Sauder
19.0	3	Edited changes and published document.	May 28, 2013	Jessica Boyle
19.0	4	Rewrote the description of source address screening (Deny Calls From Unregistered Users) for EV 184328.	June 26, 2013	Doug Sauder
19.0	4	Updated section *3.45 Transparent Proxying of SIP Headers and Options* for EV 190819.	July 5, 2013	Doug Sauder
19.0	4	Added information about the *P-Access-Network-Info* header and other *RFC 3455* headers for EV 195081.	July 12, 2013	Doug Sauder
19.0	4	Added additional information about configurable treatments and the *Reason* header for EV 196816.	July 25, 2013	Doug Sauder
19.0	4	Updated section *3.8 Offer/Answer Model* for EV 196582.	July 29, 2013	Goska Auerbach

©2019 CISCO SYSTEMS, INC.

Release	Version	Reason for Change	Date	Author
19.0	4	Removed obsolete information about the SIP MESSAGE request used for Windows Messenger instant messaging for EV 197732.	August 9, 2013	Doug Sauder
19.0	4	Edited changes and published document.	September 11, 2013	Jessica Boyle
20.0	1	Updated document for Release 20.0 features.	September 12, 2013	Doug Sauder
20.0	1	Edited changes and published document.	November 7, 2013	Joan Renaud
20.0	2	Added clarification about *Privacy:none* (EV 211138) and SIP authentication (EV 208197).	January 31, 2014	Doug Sauder
20.0	2	Corrected information about when BroadWorks sends the INFO request with stop *CallWaitingTone* for EV 206648.	March 31, 2014	Doug Sauder
20.0	2	Edited changes and published document.	May 9, 2014	Jessica Boyle
21.0	1	Updated document for Release 21.0 features.	November 3, 2014	Doug Sauder
21.0	1	Updated the BroadSoft 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 BroadWorks applies “history” privacy to the *Diversion* header.	February 27, 2015	Doug Sauder
21.0	2	Added rebranded server icons. Edited changes and published document.	March 9, 2015	Joan Renaud
21.0	3	Corrected information about emergency calls (PR-47551) and originator identification (PR-47013).	August 27, 2015	Doug Sauder
21.0	3	Changed *networkSendHistoryInfo* to *useHistoryInfoOnNetworkSide* (PR-49406).	January 29, 2016	Doug Sauder
21.0	3	Edited changes and published document.	February 5, 2016	Jessica Boyle
22.0	1	Updated document for Release 22.0 features.	September 22, 2016	Doug Sauder
22.0	1	Edited changes and published document.	December 7, 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

©2019 CISCO SYSTEMS, INC.

Release	Version	Reason for Change	Date	Author
23.0	1	Edited changes and published document.	November 29, 2018	Jessica Boyle
23.0	2	Completed rebranding for Cisco and republished document.	March 8, 2019	Jessica Boyle

©2019 CISCO SYSTEMS, INC.

Table of Contents

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

Table of Figures

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.

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

Completed rebranding 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 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 BroadWorks should relay provisional responses from secondary device endpoints such as Shared Call Appearance device endpoints.
- 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 BroadWorks Release 21.0 and Release 22.0.

- Depending on configuration, Cisco BroadWorks can send the "header" value in the *Privacy* header.
- Cisco BroadWorks sends a *Reason* header with "Call completed elsewhere" in CANCEL requests for most forking scenarios. Prior to Release 22.0, Cisco BroadWorks supported this functionality only for Shared Call Appearance forking scenarios.

©2019 CISCO SYSTEMS, INC.

- Depending on configuration, Cisco BroadWorks can 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 *accessForkingSupport* has a new option "singleDialogWIthUPDATEIfAllowed", which prevents Cisco BroadWorks from sending an UPDATE request to an endpoint that does not support it.
- Cisco BroadWorks may send the proprietary *push-notification* parameter in the *Call-Info* header.
- In order to facilitate services, Cisco BroadWorks may add new proprietary parameters *x-bw-phone-list-name* and *x-bw-igc* to the *History-Info* and *Diversion* headers.
- Cisco BroadWorks may send the *X-BroadWorks-Remote-Party-Info* header in SIP 200 responses, 18x responses, re-INVITE requests, and UPDATE requests. Before Release 22.0, BroadWorks sent this header only in initial INVITE requests.

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 changes were made in this document version:

- Changed section *3.51 Priority and Resource-Priority SIP Headers for Emergency Calls* to clarify the conditions under which Cisco BroadWorks proxies the *Resource-Priority* header or *Priority* header (PR-47551).
- Changed section *3.2 SIP Subscriber Identification/Addressing* to correct and clarify the steps Cisco BroadWorks takes to identify the originating user (PR-47013).
- Changed *networkSendHistoryInfo* (old name) to *useHistoryInfoOnNetworkSide* (new name) (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.

©2019 CISCO SYSTEMS, INC.

- 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*.

The following change, which is not related to changes in Release 21.0, was made in this document version:

- Corrected outdated information about system parameter configuration for multiple dialog support for EV 220679.

1.8 Changes for Release 20.0, Document Version 2

The following changes were made in this document version:

- Added a clarification that Cisco BroadWorks does not honor *Privacy:none* from an untrusted access device (EV 211138).
- Added more information about SIP authentication (EV 208197).
- Corrected information about Cisco BroadWorks sending the INFO request with stop *CallWaitingTone* (EV 206648).

1.9 Changes for Release 20.0, Document Version 1

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

- Ability for secondary Application Server to act as a stateless proxy server for enhanced Cisco BroadWorks geographical redundancy.
- New scenarios for ad hoc conferences, including REFER with Replaces to a remote party to add a participant to the conference, and REFER with BYE to boot a participant from a conference.
- Support for the conference event package (*RFC 4575*).
- New configuration option to determine whether Cisco BroadWorks should process the *History-Info* header or the *Diversion* header when it receives both headers in a SIP message.
- New configuration option in the “proxy unknown headers” functionality to allow injection of an unrecognized header from a redirecting user agent.
- Support for the Session Recording Protocol (call recording), including support for the *recordpref* and *record* SDP attributes (draft-ietf-siprec-protocol-09).
- Support for the URI http://127.0.0.1/silent in the *Alert-Info* header for silent alerting.

The following changes, which are not related to changes in Release 20.0, were made in this document version:

- Updated and rewrote section *3.4 Privacy Mechanism for the Session Initiation Protocol (SIP)/Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks (RFC 3323/RFC 3325)*, which covers privacy and the asserted identity.
- Added a list of all BroadSoft proprietary headers along with their syntax.

©2019 CISCO SYSTEMS, INC.

- General updates (newly issued RFCs or revised RFCs, latest versions of BroadWorks documents, and so on).

1.10 Changes for Release 19.0, Document Version 4

The following changes were made in this document version:

- Rewrote the description of source address screening (section *3.35 Deny Calls From Unregistered Users*) for EV 184328.
- Updated section *3.45 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.
- Updated section *3.8 Offer/Answer Model* for EV 196582.
- Removed obsolete information about the MESSAGE request used for Windows Messenger instant messaging for EV 197732.

1.11 Changes for Release 19.0, Document Version 3

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 2

This version of the document corrects the following error:

- The *maxHops* configuration parameter in the *AS_CLI/System/CallP/CallForwarding* command line interface (CLI) level should be the *defaultMaxRedirectionDepth* parameter in the *AS_CLI/SubscriberMgmt/Policy/CallProcessing/CallLimits* CLI level.

1.13 Changes for Release 19.0, Document Version 1

The following SIP access interface changes were made to this document for Release 19.0. They are the interface differences between BroadWorks Release 18.0 and Release 19.0.

- Updated information about Cisco BroadWorks' responses to a REGISTER request, including reason text that indicates lockout status.
- Added information about early media transitions (access side support for *RFC 3398*).
- Added information about SDP bandwidth modifiers (*RFC 3556*), which Cisco BroadWorks may add to prevent an access device from sending Real-Time Transport Protocol (RTP) media when Cisco BroadWorks provides Music on Hold.
- Updated information about SIP registration to cover Globally Identifiable Number (GIN) registration (for *RFC 6140*).
- Added information about Cisco BroadWorks IPv4/IPv6 dual-stack support.

©2019 CISCO SYSTEMS, INC.

1.14 Changes for Release 18.0, Document Version 1

The following SIP access interface changes were made to this document for Release 18.0. They are the interface differences between BroadWorks Release 17.0 and Release 18.0.

- Configurable transport in the *Contact* header (section *3.1.3.1 Differences between UDP and TCP Transports for SIP*).
- Configurable addition of the NOTIFY message following a REFER (sections *3.18 Session Initiation Protocol (SIP) Refer Method (RFC 3515)/SIP "Replaces" Header (RFC 3891)/SIP Referred-BY Mechanism (RFC 3892)* and *3.19 Session Initiation Protocol (SIP) Call Control Conferencing for User Agents (RFC 4579)/Framework for Conferencing with SIP (RFC 4353)*.
- Updated Short Message Service (SMS) and SIP MESSAGE support [section *3.23 Session Initiation Protocol (SIP) Extension for Instant Messaging (RFC 3428)*].
- Support for *max-fs* and *max-mbps* H.264 SDP payload format options in the SDP (section *3.34.2.4 SDP Handling – Video Streaming Enabled on Media Server*).
- Added Connected Line Identification Presentation (COLP), which is now supported in UPDATE and re-INVITE messages, see section *3.39 Connected Line Identification Presentation (COLP)*.
- Added Priority and Resource-Priority SIP Headers for emergency calls (section *3.51 Priority and Resource-Priority SIP Headers for Emergency Calls*).
- Added support for IPv6 (section *3.52 IPv6 Support*).

1.15 Changes for Release 17.0, Document Version 3

The following SIP access interface changes were made to this document for Release 17.0. They are the interface differences between BroadWorks Release 17.0 version 2 and Release 17.0 version 3.

- Added SIP timer information to section *3.1.4 SIP Timers*.

1.16 Changes for Release 17.0, Document Version 2

The following SIP access interface changes were made to this document for Release 17.0. They are the interface differences between BroadWorks Release 17.0 version 1 and Release 17.0 version 2.

- Added that Cisco BroadWorks now adds the *Reason* header to the Ring Splash CANCEL message in section *3.7.1 Priority Ringing on Device and Ring Splash*) for EV 113116.

1.17 Changes for Release 17.0, Document Version 1

The following SIP access interface changes were made to this document for Release 17.0. They are the interface differences between BroadWorks Release 16.0 and Release 17.0.

- Addition of section *3.47 Call Center Call Information*.
- Addition of section *3.48 Cisco BroadWorks Service* Control.
- Addition of section *3.34.2.3 Q.850 Protocol*.
- Addition of section *3.49 BroadSoft Proprietary Headers*.

©2019 CISCO SYSTEMS, INC.

- Added that Cisco BroadWorks now optionally adds new *P-Called-Party-ID* headers in addition to proxying them.

1.18 Changes for Release 16.0, Document Version 2

The following SIP access interface changes were made to this document for Release 16.0. They are the interface differences between BroadWorks Release 16.0 version 1 and Release 16.0 version 2.

- Updated section *1.1 .*
- Updated provisional response section to add deviations for EV 104113 and EV 104164.

1.19 Changes for Release 16.0, Document Version 1

The following SIP access interface changes were made to this document for Release 16.0. They are the interface differences between BroadWorks Release 15.sp2 and Release 16.0.

- Addition of transparent proxy of unknown SIP headers and options.
- Addition of Advice of Charge support.
- Addition of enhancements to media changes between early and established sessions.
- Addition of the *Call Completed Elsewhere* value to the *Reason* header.
- Addition of partial support of the *Join* header (*RFC 3911*) and *Replaces* header (*RFC 3891*) for the Shared Call Appearance service.
- Updated section *3.2 SIP Subscriber Identification/Addressing* for EV 95023.

1.20 Changes for Release 15.sp2

The following SIP access interface changes were made to this document for Release 15.sp2. They are the interface differences between 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.
- Support of “tgrp”, “trunk-context”, “otg”, and “dtg” for trunk group support.
- Addressed issues related to EV 95196.

1.21 Changes for Release 15.0

The following SIP access interface changes are introduced in BroadWorks Release 15. This version of the document includes the following changes:

- Codec details in Cisco BroadWorks Video Interactive Voice Response (IVR) Support
- Support of the *History-Info* header
- Support for subscription to message-summary event package (*RFC 3842*)

1.22 Changes for Release 14.sp6

The following SIP access interface changes are introduced in BroadWorks Release 14.sp6. This version of the document includes the following changes:

- Support for the *P-Called-Party-ID (PCPI)* SIP header

©2019 CISCO SYSTEMS, INC.

1.23 Changes for Release 14.sp5

The following SIP access interface changes are introduced in BroadWorks Release 14.sp5. This version of the document includes the following changes:

- Support for the *AccessCode* SIP header

1.24 Changes for Release 14.sp4

The following SIP access interface changes are introduced in BroadWorks Release 14.sp4. This version of the document includes the following changes:

- Conference ID is now distinct from the Conference uniform resource identifier (URI) for device-initiated conferences.

1.25 Changes for Release 14.sp3

The following SIP access interface changes are introduced in BroadWorks Release 14.sp3. This version of the document includes the following changes:

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

1.26 Changes for Release 14.sp2

The following SIP access interface changes are introduced in BroadWorks Release 14.sp2. This version of the document includes the following changes:

- 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.

1.27 Changes for Release 14.sp1

The following SIP access interface changes are introduced in BroadWorks Release 14.sp1. This version of the document includes the following changes:

- 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 though feature access code (FAC) dialing or implicitly for Hunt Group and Call Center redirection.
- Enhancement to the Message Summary to (optionally) send the number of saved and urgent messages in addition to new messages.
- Support for (optionally) sending 503 Service Unavailable responses during overload condition. The 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), UPDATE (and responses).
- Support for dual-tone multi-frequency (DTMF) signaling in INFO messages.

©2019 CISCO SYSTEMS, INC.

1.28 Changes for Release 14.0

The following SIP Access Interface changes are introduced in BroadWorks Release 14.0. This version of the document includes the following changes:

- Support of SIP REFER and UPDATE request digest MD5 authentication challenges on the Cisco BroadWorks Application Server. This enhancement provides enhanced security for devices communicating with the Cisco BroadWorks Application Server by ensuring that the REFER and UPDATE methods are challenged when authentication is activated.
- TCP Connection Management enhancements including explicit transport configuration. Cisco BroadWorks recommends devices to use connection reuse when using connection-oriented protocols such as 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 access 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, which 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 a T.38 FAX.

1.29 Changes for Release 13.0

The following SIP access interface changes are introduced in BroadWorks Release 13.0. This version of the document includes the following changes:

- Support of device-based conferencing via draft-ietf-sipping-cc-conferencing-07, based on the ad-hoc method described in section 5.4 of the draft.
- Enhancements to Digest MD5 authentication to allow usage of qop=auth.

1.30 Changes for Release 12.0

The following SIP Access Interface changes are introduced in BroadWorks Release 12.0. This version of the document includes the following changes:

- Addition of extended diversion reasons (for example, Call-Center, Hunt-Group, and so on).

©2019 CISCO SYSTEMS, INC.

- Use of SIP contact advancing for Application Server overloads 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 Cisco 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.
- Support of TCP. Cisco BroadWorks now provides TCP support in addition to UDP support.

1.31 Changes for Release 11.0

The following SIP Access Interface changes are introduced in BroadWorks Release 11.0. This version of the document includes the following changes:

- 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**, and **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.

1.32 Changes for Release 10.0

The following SIP access interface changes are introduced in BroadWorks Release 10.0. This version of the document includes the following changes:

- 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-parameters. 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.

©2019 CISCO SYSTEMS, INC.

- Sending of *privacy* headers for all calls instead of just for calls with restricted calling line identity. Now Cisco BroadWorks always includes the appropriate *privacy* headers for the following privacy versions: privacy-00, privacy-03, and *RFC 3323*. The *RFC 3323-Japan privacyVersion* 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.

1.33 Changes for Release 9.1

The following SIP Access interface changes are introduced in BroadWorks Release 9.1. This version of the document includes the following changes:

- Access Interface Forking enhancements to support the ability to fork and try more than one address during call setup toward devices on the access interface. This enhancement provides more configuration options for redundancy.
- Supports of incoming SIP digest MD5 authentication challenges on the Cisco BroadWorks Application Server. This enhancement provides enhanced security for devices expecting SIP requests from the Cisco BroadWorks Application Server.

1.34 Changes for Release 9.0

The following SIP Access Interface changes are introduced in BroadWorks Release 9.0. This version of the document includes the following changes:

- 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 echo 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 which make use of *RFC 3265*, including packages for enhanced shared call appearance lamp management and line seizure detection.
- Cisco BroadWorks has modified the format of the *Alert-Info* header to be compliant with *RFC 3261*. A scheme is now included in the *Alert-Info* header contents. The old *Alert-Info* header content format is shown:
  - *Alert-Info*: `<Bellcore-dr2>`

  The new *Alert-Info* header content format is shown:
  - *Alert-Info*: `<http://127.0.0.1/Bellcore-dr2>`

©2019 CISCO SYSTEMS, INC.

- Cisco BroadWorks has added support for Calling Party Identification with Call Waiting via the proprietary INFO-based flash support. This includes an additional body field in the INFO method sent by Cisco BroadWorks. An example INFO body for call waiting with calling party identification is shown in the following list. Note that when the calling name and/or number are unavailable, these fields are not populated in the body of the INFO method.
  - Play tone CallWaitingTone1
  - Calling-Name: "Rod Smith"
  - Calling-Number: 2403645137
- Cisco BroadWorks no longer attempts to audit devices on the access interface with the INFO method, when devices advertise support of the INFO method in the *Allow* header. Instead, Cisco BroadWorks uses the re-INVITE to audit these devices as the re-INVITE refreshes the session, which prevents the session from expiring when proxies, Application Layer Gateways, or other intermediary entities exist between Cisco BroadWorks and the device.
- Cisco BroadWorks has added support for draft-ietf-sipping-mwi-02.txt. Cisco BroadWorks no longer sends NOTIFYs for VMWI based on draft-mahy-sip-message-waiting-00.txt. Cisco BroadWorks has also added support for the Voice-Message portion of *draft-ietf-sipping-mwi-02.txt*. Cisco BroadWorks now provides both the Messages-Waiting and Voice-Message portion of draft-ietf-sipping-mwi-02.txt to the Cisco BroadWorks subscriber in the NOTIFY.
- Cisco BroadWorks has added support for Instant Messaging and Presence (IM&P) capable devices based on Microsoft Windows IM&P implementation.
- Cisco BroadWorks has added support for loose routing in compliance with *RFC 3261*.

©2019 CISCO SYSTEMS, INC.

2 Purpose

This document describes the interface used to communicate between the Cisco BroadWorks Application Server and partner access devices including SIP phones, SIP access gateways, SIP trunking gateways, and so on. The Cisco BroadWorks Application Server access interface, in contrast to the network interface, may provide access to devices on the public Internet and may be considered non-trusted. The protocol used by Cisco BroadWorks to communicate with access devices described in this document is the SIP. This document describes Cisco BroadWorks use of SIP to communicate with access devices. It describes 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.

©2019 CISCO SYSTEMS, INC.

3 Specifications

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

- RFC 1889: RTP: A Transport Protocol for Real-Time Applications, January 1996 (made obsolete by RFC 3550)
- RFC 1890: RTP Profile for Audio and Video Conferences with Minimal Control, January 1996 (made obsolete by RFC 3551)
- RFC 2327: SDP: Session Description Protocol, April, 1998 (made obsolete by RFC 4566)
- RFC 2806: URLs for Telephone Calls, April, 2000 (made obsolete by RFC 3966)
- RFC 2833: RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals, May 2000 (made obsolete by RFC 4733 and RFC 4734)
- 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 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 Third-Generation Partnership Project (3GPP), January 2003
- RFC 3515: The Session Initiation Protocol (SIP) Refer Method, April, 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 3556: Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth, July 2003
- RFC 3725: Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP), April, 2004

©2019 CISCO SYSTEMS, INC.

- 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 3891: The Session Initiation Protocol (SIP) "Replaces" Header, September 2004
- RFC 3892: The Session Initiation Protocol (SIP) Referred-By Mechanism, September 2004
- RFC 3911: The Session Initiation Protocol (SIP) Join Header
- 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 4235: An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP), November 2005
- RFC 4244: An Extension to the Session Initiation Protocol (SIP) for Request History Information, November 2005
- RFC 4353: A Framework for Conferencing with the Session Initiation Protocol (SIP), February 2006
- 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 4575: A Session Initiation Protocol (SIP) Event Package for Conference State, August 2006
- RFC 4579: Session Initiation Protocol (SIP) Call Control – Conferencing for User Agents, August 2006
- RFC 4629: RTP Payload Format for ITU-T Rec. H.263 Video, January 2007
- RFC 4733: RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals, December 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, September 2007
- RFC 5168: XML Schema for Media Control, March 2008
- RFC 5806: Diversion Indication in SIP, March 2010[1]

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

©2019 CISCO SYSTEMS, INC.

- RFC 5923: Connection Reuse in the Session Initiation Protocol (SIP), June 2010
- 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 6140: Registration for Multiple Phone Numbers in the Session Initiation Protocol (SIP), March 2011
- RFC 6665: SIP-Specific Event Notification, July 2012
- RFC 6947: The Session Description Protocol (SDP) Alternate Connectivity (ALTC) Attribute, May 2013
- RFC 7315: Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3GPP, July 2014
- draft-ietf-siprec-protocol-09.txt: Session Recording Protocol, October 2013
- 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
- ITU-T T.38 Procedures for Real-time Group 3 Facsimile Communication over IP Networks, September 2005
- 3GPP TS 24.647 v8.0.0 Advice of Charge (AoC)
- GSMA IR.94 IMS Profile for Conversational Video Service, Version 10.0. October 2015

©2019 CISCO SYSTEMS, INC.

3.1 Session Initiation Protocol (RFC 3261)

Cisco BroadWorks supports all standard SIP functionality. Following are highlights and clarifications of this support:

- Cisco BroadWorks is implemented as a back-to-back user agent (B2BUA), 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 can be configured to support IPv4 only, IPv6 only, or both IPv4 and IPv6 simultaneously.
- 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 by default. However, specific access devices may be configured as trusted devices.
- 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 Authentication

Cisco BroadWorks supports initiating challenges as well as responding to challenges from devices on the access interface. Cisco BroadWorks supports sending and receiving SIP digest MD5 authentication challenges only.

The following clarifications are provided for Cisco BroadWorks authentication support:

- If an access device has authentication disabled, then Cisco BroadWorks does not enforce authentication for requests from that device. Authentication for an access device is enabled or disabled as an option on the access device type ("Identity/Device Profile Type" in the web interface).
- If a user[2] does not have the Authentication service assigned, then Cisco BroadWorks does not enforce authentication of any SIP requests from that user.
- The Cisco BroadWorks Application Server enforces authentication of REGISTER, INVITE, MESSAGE, REFER, UPDATE, and SUBSCRIBE requests, subject to the following conditions:

[2] If the user is a trunking user, and if the call originates via a trunk group, then the user does not need to have the Authentication service assigned. In this case, BroadWorks enforces authentication according to the configuration of the trunk group. However, if a trunking user originates a call from a secondary device, such as a Shared Call Appearance device, then BroadWorks does require the Authentication service. For details, see the *BroadWorks SIP Trunking Solution Guide* [66].

©2019 CISCO SYSTEMS, INC.

- Cisco BroadWorks enforces authentication of REGISTER requests, SUBSCRIBE requests, and MESSAGE requests without any special considerations.
- If an access device has the *Authenticate REFER* option enabled, then Cisco BroadWorks enforces authentication of REFER requests from that device.
- Cisco BroadWorks challenges INVITE requests according to the value of the SIP parameter *inviteAuthenticationRatio*. If the parameter's value is "0", then Cisco BroadWorks does not challenge INVITE requests. If the parameter's value is "1", then Cisco BroadWorks challenges every initial INVITE request (and optionally re-INVITE requests, as described in the following point). If the parameter's value is between "0" and "1", then Cisco BroadWorks challenges the specified fraction of INVITE requests. The default value for *inviteAuthenticationRatio* is "0".
- To clarify the preceding point: Cisco BroadWorks applies the parameter *inviteAuthenticationRatio* to challenge initial INVITE requests and optionally re-INVITE requests and certain UPDATE requests. If the SIP parameter *reInviteAuthentication* is set to "false", then Cisco BroadWorks challenges only initial INVITE requests. If the SIP parameter *reInviteAuthentication* is set to "true", then Cisco BroadWorks challenges initial INVITE requests, re-INVITE requests, and qualifying UPDATE requests. An UPDATE request is "qualifying" if it contains an SDP or if it changes the contact URI. The default value for *reInviteAuthentication* is "true".
- If Cisco BroadWorks challenges an INVITE request or an UPDATE request and the subsequent authentication attempt fails, then Cisco BroadWorks behaves as if *inviteAuthenticationRatio* were temporarily set to "1" for that device endpoint. (This means it challenges every INVITE request from that device endpoint.) If a subsequent authentication succeeds, then Cisco BroadWorks resumes its normal behavior for that device endpoint.

- The source of the user name and password for authentication depend on the access device configuration. Moreover, when Cisco BroadWorks issues an authentication challenge, the realm source also depends on the access device configuration. This information is described in the following table for users who are not SIP trunking users. For SIP trunking users, see the *Cisco BroadWorks SIP Trunking Solution Guide* [66].

Device Type Authentication Option		
Field	**Enabled**	**Enabled With Web Portal Credentials**
User name	Provisioned within the user's Authentication Service	The user's BroadWorks user identity. This is the identity the user uses to log in to the Web Portal or an Xsi application.
Password	Provisioned within the user's Authentication Service	A 40-character string generated from the user's Cisco BroadWorks password (which is the password the user uses to log in to the Web Portal or an Xsi application). To generate the 40-character string, the Application Server computes the 160-bit SHA-1 digest of the password, and then converts the digest to a hexadecimal string in which all letters are in lower case. For example, if the user's password is "secret", then the 40-character string is "e5e9fa1ba31ecd1ae84f75caaa474f3a663f05f4".

©2019 CISCO SYSTEMS, INC.

Device Type Authentication Option		
Realm	If the SIP parameter *useDomainForRealm* is set to “false” (the default value), then Cisco BroadWorks selects the realm to be the value of the SIP parameter *defaultRealm*, which is set to “BroadWorks” and cannot be changed. If the *useDomainForRealm* is set to “true”, then Cisco BroadWorks selects the realm to be the user’s call processing domain.	The domain part of the user’s Cisco BroadWorks identity. If the system is configured such that user identities have no domain part, then the realm is the system domain.

- The Cisco BroadWorks Application Server does not support MD5-session or qop auth-int.
- The Cisco BroadWorks Application Server does not authenticate responses and does not generate the *Authentication-Info* header to allow responses to be authenticated.
- The next nonce of the *Authentication-Info* header is only supported within the associated dialog. The value is not shared with the redundant Cisco BroadWorks Application Server. This potentially may cause extra authentication challenges.
- The Cisco BroadWorks Application Server reuses a nonce within a dialog until newly challenged. The nonce is not reused outside of the associated dialog unless it relates to the specific challenge. This potentially may cause extra authentication challenges.
- The Cisco BroadWorks Application Server is unable to honor more than one authentication challenge within 401 and 407 responses.
- If the Application Server receives a request with an expected Authorization realm and rejects the request because of an authentication failure, then the Application Server sends a 403 response instead of a 401 response.
- A series of authentication failures can cause the Application Server to “lock out” a device endpoint. When the device endpoint is locked out, the Application Server immediately sends a 403 response, without attempting to authenticate the request.
- Because the 403 response may be sent for various reasons, the Application Server sets the status-line reason-phrase to provide additional information about the authentication failure and the lockout status of the device endpoint:
  - “403 Authentication Failure” indicates that the request failed authentication.
  - “403 Authentication Failure Lockout” indicates that the request failed authentication and caused a lockout of the device endpoint or trunk group.
  - “403 Locked Out” indicates that the request failed because the device endpoint or trunk group is locked out.
  - “403 Authentication Loop” indicates that the request failed because the Application Server considered it part of an authentication loop.

3.1.2 Support of the OPTIONS Method

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

©2019 CISCO SYSTEMS, INC.

Access devices must support receiving the OPTIONS request. The access device must provide a SIP response to the OPTIONS request. However, the access 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.

3.1.3 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 which transport to use and how to handle failures. The following sections provide clarification on Cisco BroadWorks TCP implementation.

3.1.3.1 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 Transport Layer Security (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 NAPTR lookup is performed for the target. A NAPTR service field of "SIP+D2T" indicates TCP and "SIP+D2U" indicates UDP. The preference and order of the NAPTR record is 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, an 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 round, 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.

©2019 CISCO SYSTEMS, INC.

- When TCP transport is used, Cisco BroadWorks 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 (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 reflected by NAPTR, if used, and the transport desired by the other device to be disregarded.
  - Cisco BroadWorks includes the transport parameter within the SIP URI *Request-URI* header and SIP URI *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 the *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**>

3.1.3.2 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 will not re-use the existing connection causing a connection to be set up in each direction for the call.

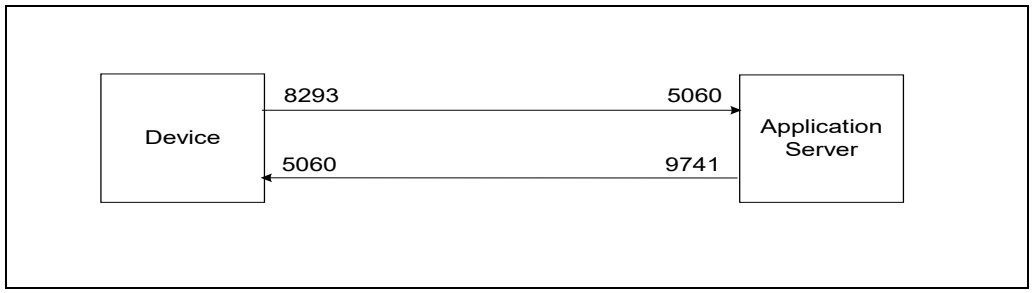

Figure 1 TCP Connection Management

©2019 CISCO SYSTEMS, INC.

(R-1) Cisco BroadWorks does not support *RFC 5923*. However, Cisco BroadWorks recommends devices that support TCP to reuse the existing connections within a dialog.

(R-2) It is recommended that all devices support the following requirements when utilizing the TCP transport with Cisco BroadWorks.

(R-3) The device must not actively close any sockets unless:

(R-3) a. An error is encountered on the socket.

(R-3) b. A connection has become stale (determined by configuration on the device).

(R-3) c. The maximum number of SIP/TCP sockets supported on the device has been reached, and resources must be reclaimed.

(R-3) d. The maximum number of SIP/TCP sockets per peer supported on the device has been reached, and resources must be reclaimed.

(R-3) e. The device is undergoing maintenance requiring the sockets to be terminated.

(R-4) The device should reuse sockets whenever possible:

(R-4) a. All SIP responses must be sent on the socket of the corresponding Request.

(R-4) b. After resolving the destination IP address and port, but before sending any SIP request, the device should determine if 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.

3.1.4 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	This is not used by Cisco BroadWorks (see Timer I and Timer K in this table).
Timer A	Timer A is used by Cisco BroadWorks as defined in *RFC 3261*. It is initialized to T1 and doubles at every retransmission. ▪ When Cisco BroadWorks is in Yellow CallP Overload condition, Timer A is initialized to 2*T1. ▪ When Cisco BroadWorks is in Red CallP Overload condition, Timer A is initialized to 4*T1.

©2019 CISCO SYSTEMS, INC.

Timer	Comment
Timer B	Timer B is used by Cisco BroadWorks as defined in *RFC 3261* for Transmission Control Protocol (TCP). It is initialized to 64*T1. For User Datagram Protocol (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. BroadWorks can route advance a request to multiple IP address and port locations according to *RFC 3263*. In this situation, BroadWorks may use a shorter Timer B value for all locations except for the last, based on the *suspiciousAddressThreshold* configuration (if configured). ▪ When the transport is UDP, the *suspiciousAddressThreshold* is the number of delivery attempts before advancing to the next location. ▪ When the transport is TCP, the *suspiciousAddressThreshold* is used to compute the value of Timer B as follows: ▪ If *suspiciousAddressThreshold* is 1, the value is "T1". ▪ If *suspiciousAddressThreshold* is n, the value is the one for (n – 1) plus the minimum between T2 and 2^(n-1)*T1 (similar to the non-INVITE retry rule defined in *RFC 3261*). ▪ When Cisco BroadWorks is in Yellow CallP Overload condition, an extra 2 seconds are added. ▪ When Cisco BroadWorks is in Red CallP Overload condition, an extra 4 seconds are added. Regardless of the above, the maximum value is 32 seconds. Note that the same logic applies to Timer F as well.
Timer C	Timer C is not used by Cisco BroadWorks because it does not act as a stateful proxy but rather as a B2BUA instead.
Timer D	The wait time for late retransmission is handled differently in Cisco BroadWorks. Received messages are remembered by Cisco BroadWorks and kept in memory until the stale message audit is run. 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 *useSessionCompletionTimer* is true, the dialog post-released retention period is configured using *sessionCompletionTimer* value (5 seconds to 100 seconds). ▪ If *useSessionCompletionTimer* is false, the dialog post-released retention period is set to "10*T2". When Cisco BroadWorks is in Yellow or Red CallP Overload condition, dialog post-released retention is forced to 5 seconds instead. Note that the same logic applies to Timers I, J, and K.
Timer E	Timer E is used by Cisco BroadWorks as defined in *RFC 3261*. It is initialized to T1 and doubles at every retransmission until beginning to use T2 as the interval. Cisco BroadWorks does not start using interval T2 upon receiving a 1xx response; the *RFC 3261* non-compliance is to accommodate devices that are not compliant with *RFC 4320*. ▪ When Cisco BroadWorks is in Yellow CallP Overload condition, Timer E is initialized to 2*T1. ▪ When Cisco BroadWorks is in Red CallP Overload condition, Timer E is initialized to 4*T1.

©2019 CISCO SYSTEMS, INC.

Timer	Comment
Timer F	Timer F is used by Cisco BroadWorks 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 Timer B.
Timer G	Timer G is used by Cisco BroadWorks as defined in *RFC 3261*. It is initialized to T1 and doubles 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. ▪ When Cisco BroadWorks is in Yellow CallP Overload condition, Timer G is initialized to 2*T1. ▪ When Cisco BroadWorks is in Red CallP Overload condition, Timer G is initialized to 4*T1.
Timer H	Timer H is used by Cisco BroadWorks as defined in *RFC 3261* for TCP (unless retrying a response as described for Timer G). It is initialized 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 comment for Timer D.
Timer J	The wait time for late retransmission is handled differently in Cisco BroadWorks. See comment for Timer D.
Timer K	The wait time for late retransmission is handled differently in Cisco BroadWorks. See comment for Timer D.

3.1.5 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 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.

©2019 CISCO SYSTEMS, INC.

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.

3.1.6 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.

3.1.6.1 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
```

3.1.6.2 P-Charging-Vector Impact

Cisco BroadWorks builds the *P-Charging-Vector* header using the Application Server IP address or fully qualified domain name (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.

©2019 CISCO SYSTEMS, INC.

3.1.7 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 it builds the *branch* parameter value using the default format previously described.

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.

3.1.7.1 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 failure.

3.1.7.2 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.

©2019 CISCO SYSTEMS, INC.

3.1.8 Call-Info Header

Cisco BroadWorks supports the *Call-Info* header for specialized purposes, particularly for sharing information with multiple access devices in a Shared Call Appearance arrangement. Cisco BroadWorks sends and receives the *Call-Info* header in INVITE requests. Cisco BroadWorks also receives the *Call-Info* header in SUBSCRIBE requests and sends the *Call-Info* header in NOTIFY requests for various event packages. This functionality is described in detail in the *BroadWorks Shared Call Appearance Interface Specification* [67] and the *BroadWorks SIP Access Side Extensions Interface Specification Guide* [43].

Cisco BroadWorks may send the *Call-Info* header in INVITE requests to support push notifications to access devices. The following is an example of an INVITE request with a push notification.

```
INVITE sip:9725551212@10.16.134.100 SIP/2.0
Via:SIP/2.0/UDP 10.16.134.35;branch=z9hG4bKBroadWorks.1lp0lio-
10.16.134.100V5060-0-1001438685-1021370961-1447259372473-
From:"User04
North_as90"<sip:9725551234@broadsoft.com;user=phone>;tag=1021370961-
1447259372473-
To:"User02 North_as90"<sip:9725551212@broadsoft.com>
Call-ID:BW112932473111115391435356@10.16.134.35
CSeq:1001438685 INVITE
Contact:<sip:10.16.134.35:5060>
Supported:100rel
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Call-Info:<sip:broadsoft.com>;push-notification=41335%230
Recv-Info:x-broadworks-client-session-info
Accept:application/btbc-session-info,application/dtmf-
relay,application/media_control+xml,application/sdp,multipart/mixed
Max-Forwards:10
Content-Type:application/sdp
Content-Length:339

v=0
o=BroadWorks 16 1 IN IP4 10.16.134.102
s=-
c=IN IP4 10.16.134.102
b=AS:512
t=0 0
a=sendrecv
m=audio 2278 RTP/AVP 9 102 0 8 18 127
a=rtpmap:9 G722/8000
a=rtpmap:102 G7221/16000
a=fmtp:102 bitrate=32000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:127 telephone-event/8000
```


©2019 CISCO SYSTEMS, INC.

3.2 SIP Subscriber Identification/Addressing

SIP subscriber identification/addressing:

- URLs for Telephone Calls (*RFC 2806*)
- The tel URI for Telephone Numbers (*RFC 3966*)
- 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*)
- 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. Highlights and clarifications of this support are as follows:

- Addressing Information:
  - Cisco BroadWorks can (optionally) restrict the calling party identity upon receipt of anonymous in the display name of the *From* header. When Cisco BroadWorks receives “anonymous” in the display name of the *From* header from an access device, Cisco BroadWorks treats the calling party identity as restricted and does not pass the calling party identity out to any untrusted entities. Cisco BroadWorks also supports other privacy specifications described in the following list on the access interface, which can also restrict the calling party identity.
  - For calls with restricted calling party identity to be sent to an access device from Cisco BroadWorks, Cisco BroadWorks inserts "Anonymous" <sip:anonymous@anonymous.invalid> in the *From* header to honor the calling party identity blocking request.
  - Cisco BroadWorks also 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.
  - To identify the originating user from a new INVITE request, Cisco BroadWorks performs the following steps:
    1) If the request has a *P-Preferred-Identity* header, then Cisco BroadWorks attempts a user lookup using the URI from that header. If the lookup succeeds, then Cisco BroadWorks selects the matched user.
    2) Otherwise, if the request has a *P-Asserted-Identity* header, then Cisco BroadWorks attempts a user lookup using the URI from that header. If the lookup succeeds, then Cisco BroadWorks selects the matched user.
    3) Otherwise, if the request has a *Remote-Party-ID* header, then Cisco BroadWorks attempts a user lookup using the URI from that header. If the lookup succeeds, then Cisco BroadWorks selects the matched user.
    4) Otherwise, if the request has no *Remote-Party-ID* header, then Cisco BroadWorks attempts a user lookup using the URI from the *From* header. (The *From* header is mandatory, per *RFC 3261*.) If the lookup succeeds, then Cisco BroadWorks selects the matched user.

©2019 CISCO SYSTEMS, INC.

NOTE: If the request has a *Remote-Party-ID* header and the lookup using the *Remote-Party-ID* URI fails, then the user lookup procedure fails. In other words, Cisco BroadWorks attempts a user lookup using either the *Remote-Party-ID* URI or the *From* URI, but not both, and the *Remote-Party-ID* URI has precedence.

- For INVITES sent from a Cisco BroadWorks server:

Header	Format	Parameter	Value
Request-URI	SIP URI	*User*	Access device user name/address of record/line/port.
		Host	Access device domain name or IP address as specified in device inventory or registered contact (for devices that register).
From		*display-name*	User or group name, if available.
	SIP URI	*User*	User or group phone number. Number is in national number format.
		Host	Application Server cluster domain name, Subscriber domain name, or IP address.
To	SIP URI	*User*	Access device user name/address of record/line/port.
		Host	Access device domain name or IP address as specified in device inventory or registered contact (for devices that register).
Contact	SIP URI	*User*	Not used (empty).
		Host	Application Server cluster domain name or IP address.

- For INVITES sent from an Access 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	Application Server cluster domain name, Subscriber domain name, Application Server alias, or IP address.
	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*	Access device user name/address of record/line/port.
		Host	Access device domain name or IP address or Application Server cluster domain name, Subscriber domain name, Application Server alias, or Application Server IP address (for devices that register).

©2019 CISCO SYSTEMS, INC.

Header	Format	Parameter	Value
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	Application Server cluster domain name, Subscriber domain name, Application Server alias, or IP address.
	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	Access device domain name or IP address.
Remote-Party-ID		*display-name*	Calling party name, if available.
	SIP URI	*User*	Access device user name/address of record/line/port.
		Host	Access device domain name or IP Address or Application Server cluster domain name, Subscriber domain name, Application Server alias, or Application Server IP address (for devices that register).
P-Asserted-Identity		*display-name*	Calling party name, if available.
	SIP URI	*User*	Access device user name/address of record/line/port.
		Host	Access device domain name, or IP Address, or Application Server cluster domain name, Subscriber domain name, Application Server alias, or Application Server IP address (for devices that register).
P-Preferred-Identity		*display-name*	Calling party name, if available.
	SIP URI	*User*	Access device user name/address of record/line/port.
		Host	Access device domain name, or IP Address, or Application Server cluster domain name, Subscriber domain name, Application Server alias, or Application Server IP address (for devices that register).

©2019 CISCO SYSTEMS, INC.

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

Cisco BroadWorks fully supports the tel URI scheme, as specified in *RFC 2806*. An access device may use either a tel URI or a SIP URI in requests sent to Cisco BroadWorks. SIP URI is the recommended format for access devices to use when sending messages to Cisco BroadWorks.

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

©2019 CISCO SYSTEMS, INC.

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

Cisco BroadWorks supports *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 supports *RFC 3323* by default, but may be alternatively configured to support the privacy features of older Internet drafts by setting the SIP parameter *privacyVersion* The information presented in this section depends on Cisco BroadWorks being configured to support *RFC 3323*.

The provisions in *RFC 3325* depend on the concept of a "trust domain". In Cisco BroadWorks, there is no formal notion of a trust domain. However, Cisco BroadWorks does make a clear distinction between trusted devices and untrusted devices. All network devices are trusted. Any particular access device may or may not be trusted, depending on the configuration of the access device.

RFC 3325 describes two SIP headers – *P-Asserted-Identity* and *P-Preferred-Identity*. In Cisco BroadWorks, these headers supersede any other headers when it comes to matching a user, as explained in section *3.2 SIP Subscriber Identification/Addressing*. *RFC 3325* specifies that a party in a call dialog can be assigned one or more network-asserted identities within a trust domain indicated by one or more *P-Asserted-Identity* headers.

Although multiple *P-Asserted-Identity* headers are allowed within a SIP message, 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 asserted identity for a given user in a SIP message. This identity includes a URI and may also include a display name.

Cisco BroadWorks acts as a privacy service as described in *RFC 3323*. *RFC 3323* defines the *Privacy* header, which can contain 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. *RFC 3325* defines "id" as an additional *Privacy* header value, and *RFC 4244* defines "history".

Cisco BroadWorks supports privacy as an originating service that can be configured by the user or invoked on a per-call basis. Accordingly, Cisco BroadWorks interprets the *Privacy* header in the context of this service. This means, for example, that if Cisco BroadWorks receives a *Privacy* header with the value "user", it interprets this as a request to apply anonymous presentation for the originating user. While Cisco BroadWorks generally processes the *Privacy* header as one would expect, there are two special situations to note:

- Cisco BroadWorks supports feature access codes (FACs) to enable or disable caller identity blocking (that is, anonymous presentation) for the current call. If the *Request-URI* of the incoming INVITE request has one of these FACs, then Cisco BroadWorks applies anonymous presentation, or not, depending on the FAC, regardless of the value of the *Privacy* header. In short, these FACs have the highest precedence.

©2019 CISCO SYSTEMS, INC.

- Whether Cisco BroadWorks acts on a *Privacy* header with the value "none" depends on the setting of the SIP parameter *supportPrivacyNone* and on whether the access device is trusted. If *supportPrivacyNone* is set to "false", or if the access device is untrusted, then Cisco BroadWorks ignores the value "none" in the *Privacy* header. On the other hand, if the value is "true" and the access device is trusted[3], then Cisco BroadWorks acts on the value "none", such that it disables anonymous presentation if it is enabled in the originating service profile. (However, as noted in the previous point, the FAC has higher precedence than the *Privacy* header.) The default value for *supportPrivacyNone* is "false".

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

- "none": This indicates that the privacy service must not add privacy. In terms of Cisco BroadWorks, this means that calling line identity (CLID) blocking should be disabled and the identity of the originator should be made available. Cisco BroadWorks may or may not honor this request from the access device, as previously explained.
- "critical": This 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, then Cisco BroadWorks rejects the INVITE request with a 500 (Server Error) response as required by *RFC 3323*. In particular, Cisco BroadWorks cannot enforce privacy of type "session"; therefore, if it receives "session" and "critical" in the *Privacy* header, it rejects the request.
- "user": This 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 this value request as requesting anonymous presentation.
- "header": This 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*, *Record-Route*, and *Contact* headers. Cisco BroadWorks provides this capability by default as a back-to-back User Agent.

  Moreover, if an INVITE request has a *History-Info* header or a *Diversion* header, then Cisco BroadWorks also applies the privacy protection to that header, as described in section *3.6 Cisco BroadWorks Support for Request History*.

  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": This indicates that the information held in the session description (SDP) should be hidden. To accomplish this, a privacy service would have to terminate the session on one end and originate one on the other end. Since Cisco BroadWorks does not support this functionality, if Cisco BroadWorks receives an INVITE request with a *Privacy* header that indicates both "session" and "critical", it rejects the request with a 500 (Server Error) response.

[3] In IMS mode, it is sufficient that *supportPrivacyNone* be set to "true" and it is not required that the access device be trusted.

©2019 CISCO SYSTEMS, INC.

To satisfy *RFC 7044*, if an INVITE request has a *History-Info* header, then Cisco BroadWorks also applies the privacy protection to that header, as described in section *3.6 Cisco BroadWorks Support for Request History*. Though not required by *RFC 7044*, Cisco BroadWorks also applies privacy protection to the *Diversion* header in a similar way.

- "id": This indicates that the network should protect the network asserted identity from leaving the Trust Domain. Cisco BroadWorks treats this privacy request as requesting anonymous presentation.
- "history": 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 *3.6 Cisco BroadWorks Support for Request History*. Cisco BroadWorks also applies "history" privacy to the *Diversion* header.

NOTE: Cisco BroadWorks does not act on a *Privacy* header in a REFER request or a 302 response, unless it interprets the 302 response as a PBX redirection. For such a PBX redirection, Cisco BroadWorks applies "history" privacy to the *History-Info* header or *Diversion* header as described in section *3.6.3 Receiving Request History*.

If Cisco BroadWorks sends an INVITE request to an access device for a call with anonymous presentation, the header values in that request depend on whether the access device is trusted. For a trusted access device, Cisco BroadWorks adds a *P-Asserted-Identity* header with the identity of the caller and adds a *Privacy* header with the values "id" and "critical". If the SIP parameter *includePrivacyUser* is "true", then Cisco BroadWorks also adds the value "user" to the *Privacy* header. The value of the *From* header depends on the value of the SIP parameter *encryptFromHeader*. If the parameter is set to "false", then Cisco BroadWorks sets the display name to the value of the SIP parameter *restrictedDisplayName* but does not change the URI to protect privacy. However, if the parameter is set to "true", then Cisco BroadWorks omits the display name and sets the URI to "sip:anonymous@anonymous.invalid".

For an untrusted access device, Cisco BroadWorks omits both the *Privacy* header and the *P-Asserted-Identity* header. Furthermore, Cisco BroadWorks makes the *From* header anonymous, setting the display name to the value of the SIP parameter *restrictedDisplayName* and setting the URI to "sip:anonymous@anonymous.invalid".

Cisco BroadWorks never populates an outgoing message with a *P-Preferred-Identity* header. Moreover, Cisco BroadWorks sends the *P-Asserted-Identity* header in INVITE requests only to trusted access devices. Cisco BroadWorks may send the *P-Asserted-Identity* header in re-INVITE requests, UPDATE requests, or responses to INVITE requests to trusted access devices to relay the connected identity. For more information on the connected identity, see *3.39 Connected Line Identification Presentation (COLP)*.

©2019 CISCO SYSTEMS, INC.

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

Cisco BroadWorks also 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 does not use either of these drafts when sending messages on the access interface. Since the access interface is considered untrusted, Cisco BroadWorks only populates the *From* header and does not include the *Remote-Party-ID* headers.

Cisco BroadWorks accepts messages on the access interface with Remote-Party-ID headers. The *Remote-Party-ID header*, when present, takes precedence over the *From* header and Cisco BroadWorks uses the *Remote-Party-ID* header to identify the Cisco BroadWorks subscriber when present.

As stated in the previous section, when calling party information is restricted, Cisco BroadWorks sends a *From* header in the INVITE with a Name-Addr as shown in the following example.

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


©2019 CISCO SYSTEMS, INC.

3.6 Cisco BroadWorks Support for Request History

3.6.1 Processing Model

Cisco BroadWorks supports the *History-Info* header and the *Diversion* header, as well as the 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's 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

©2019 CISCO SYSTEMS, INC.

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

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

©2019 CISCO SYSTEMS, INC.

3.6.2 Compliance

3.6.2.1 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 Internet Engineering Task Force (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".)

©2019 CISCO SYSTEMS, INC.

- 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 the 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
```


©2019 CISCO SYSTEMS, INC.

```
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"
```

3.6.2.2 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.

©2019 CISCO SYSTEMS, INC.

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 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 /
```


©2019 CISCO SYSTEMS, INC.

```
    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"
```

3.6.2.3 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”

©2019 CISCO SYSTEMS, INC.

- 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*.

3.6.3 Receiving Request History

3.6.3.1 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 *3.6.3.3 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 *3.6.3.4 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 two entries' SIP URIs and asserts a match if the URI user parts match and URI host parts match.

3.6.3.2 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.

©2019 CISCO SYSTEMS, INC.

- 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 the above conditions are satisfied, Cisco BroadWorks removes the header entry.

3.6.3.3 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 *3.6.3.2 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.

©2019 CISCO SYSTEMS, INC.

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.

3.6.3.4 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 *3.6.3.2 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

©2019 CISCO SYSTEMS, INC.

Internal Representation	Syntax 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.
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 if 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.

3.6.4 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.

©2019 CISCO SYSTEMS, INC.

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.

3.6.4.1 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

©2019 CISCO SYSTEMS, INC.

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.

©2019 CISCO SYSTEMS, INC.

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 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 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

©2019 CISCO SYSTEMS, INC.

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 as follows. 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

3.6.4.2 Sending Diversion Header

When Cisco BroadWorks has request history information to send in an outgoing request, it sends the *Diversion* header if it is 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".

©2019 CISCO SYSTEMS, INC.

- 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.

3.6.5 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, then 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:

©2019 CISCO SYSTEMS, INC.

- 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.

3.6.6 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.

3.6.6.1 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, Cisco BroadWorks adds a new entry and the total redirection count of all *Diversion* header entries may exceed the configured *defaultMaxRedirectionDepth* value.

The 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
```


©2019 CISCO SYSTEMS, INC.

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.

3.6.6.2 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".

3.6.7 Call Flows

3.6.7.1 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 only preserves 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.

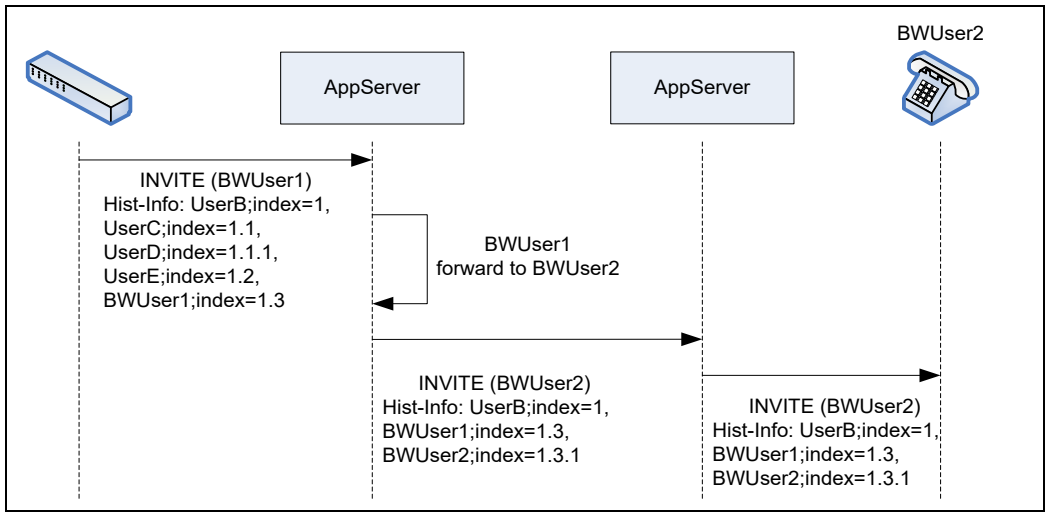

Figure 2 History-Info Indexing

©2019 CISCO SYSTEMS, INC.

3.6.7.2 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:

- 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.*

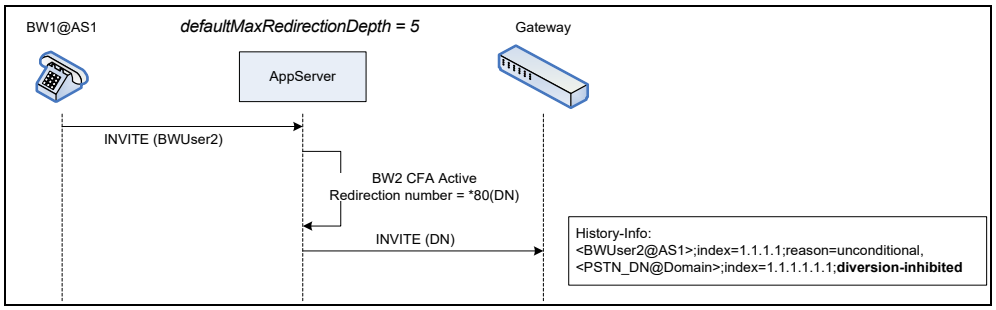

Figure 3 Diversion Inhibited with Redirection (History-Info Header)

- 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".

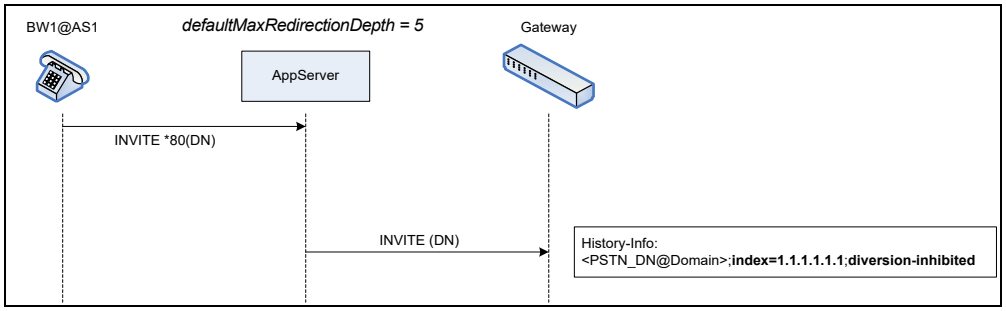

Figure 4 Diversion Inhibited on Origination (History-Info Header)

3.6.7.3 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 "anonymous@anonymous.invalid"). Also, Cisco BroadWorks proxies a received *Privacy history* and *History-Info* header from a trusted entity to a trusted entity, if no internal redirection occurs. 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 message flow when Cisco BroadWorks receives an INVITE request with the *Privacy* header set to "history". As shown, 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.

©2019 CISCO SYSTEMS, INC.

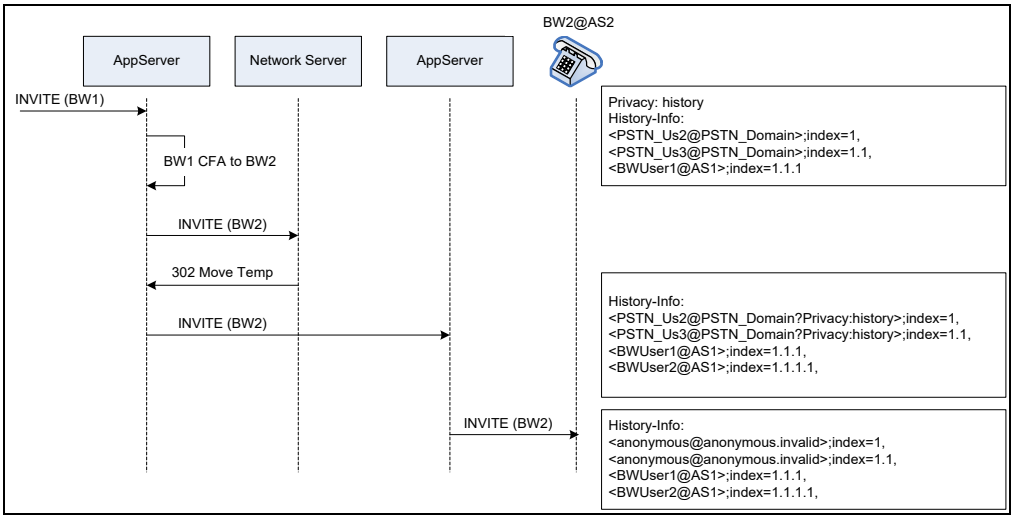

Figure 5 History-Info and Privacy Header

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

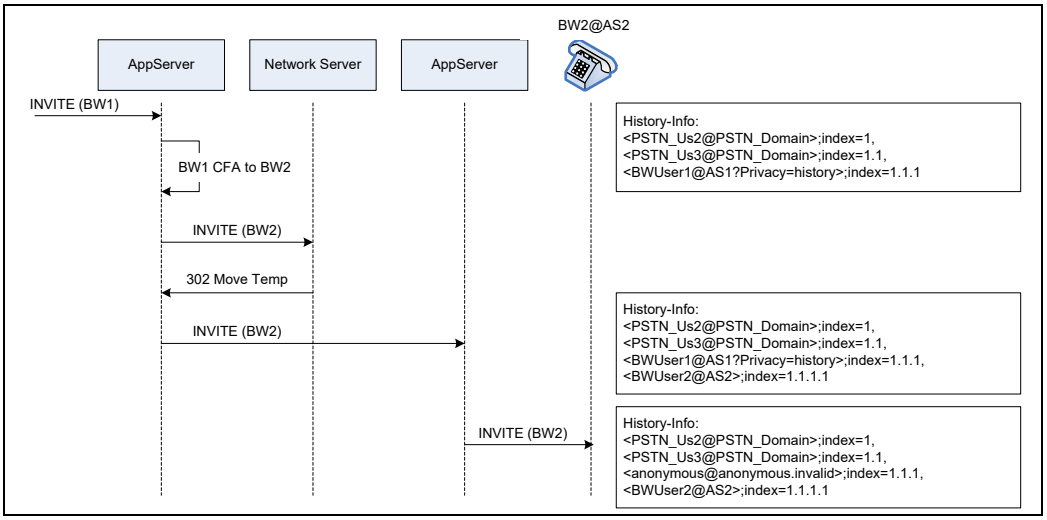

Figure 6 History-Info and Privacy Attributes

3.6.7.4 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.

©2019 CISCO SYSTEMS, INC.

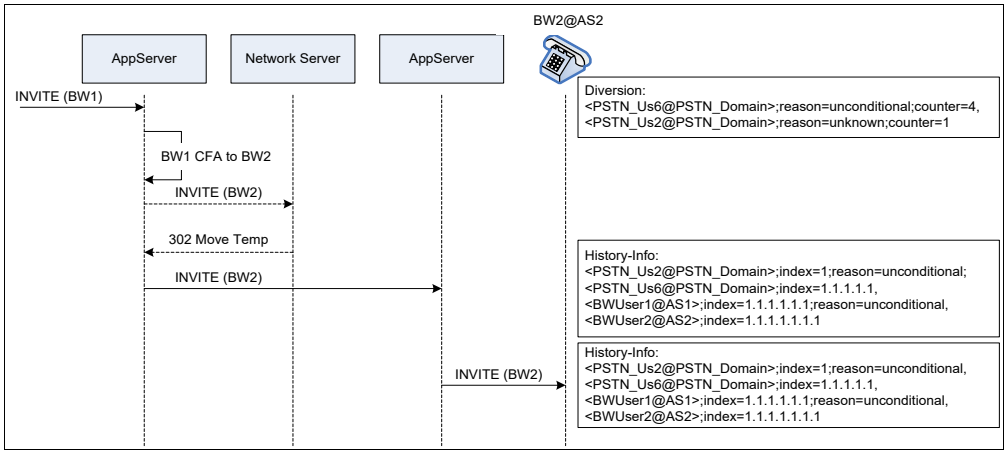

Figure 7 Counter to Index Conversion

3.6.7.5 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.

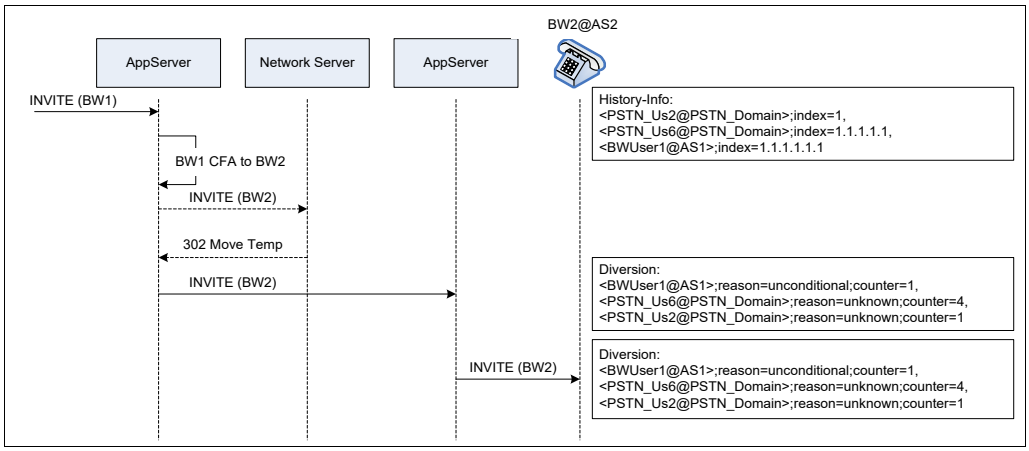

Figure 8 Index to Counter Conversion

3.6.7.6 History-Info with Cause Parameter

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

©2019 CISCO SYSTEMS, INC.

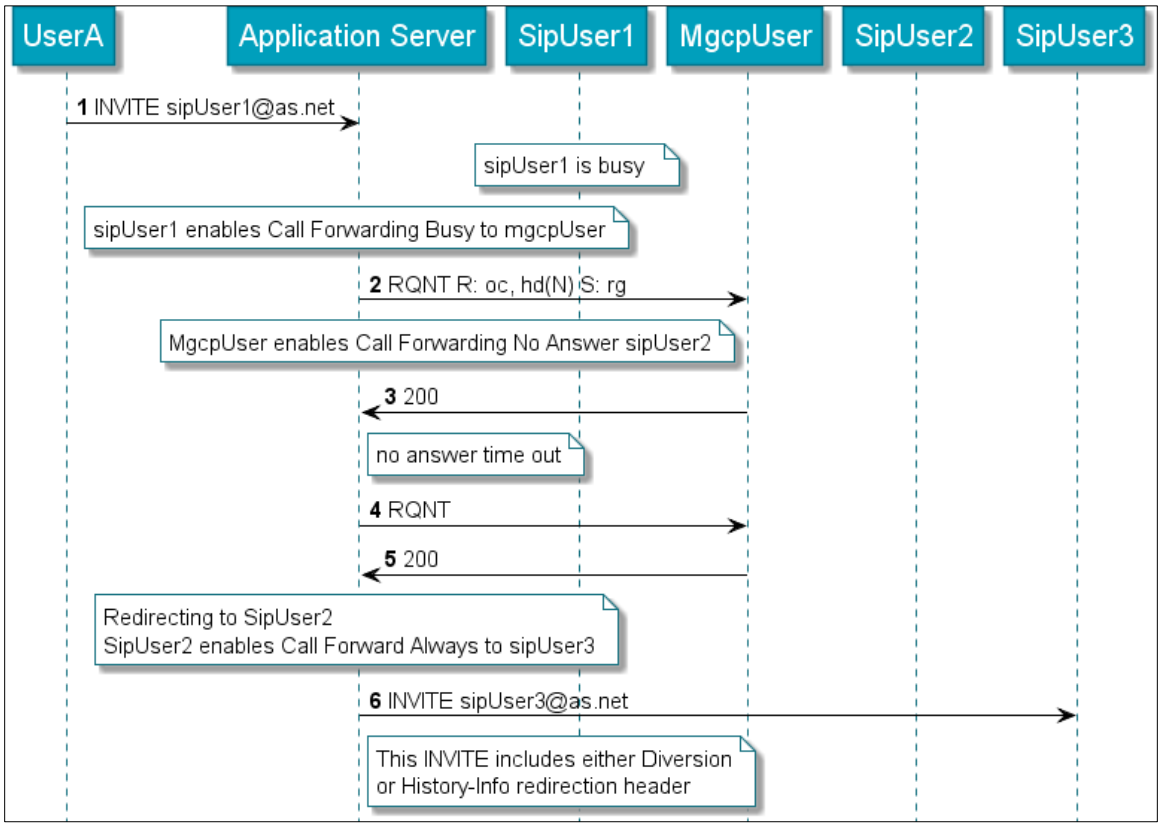

Figure 9 Cisco BroadWorks User Diversion Call Flow

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*.

©2019 CISCO SYSTEMS, INC.

```
<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
```

3.6.7.7 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:

©2019 CISCO SYSTEMS, INC.

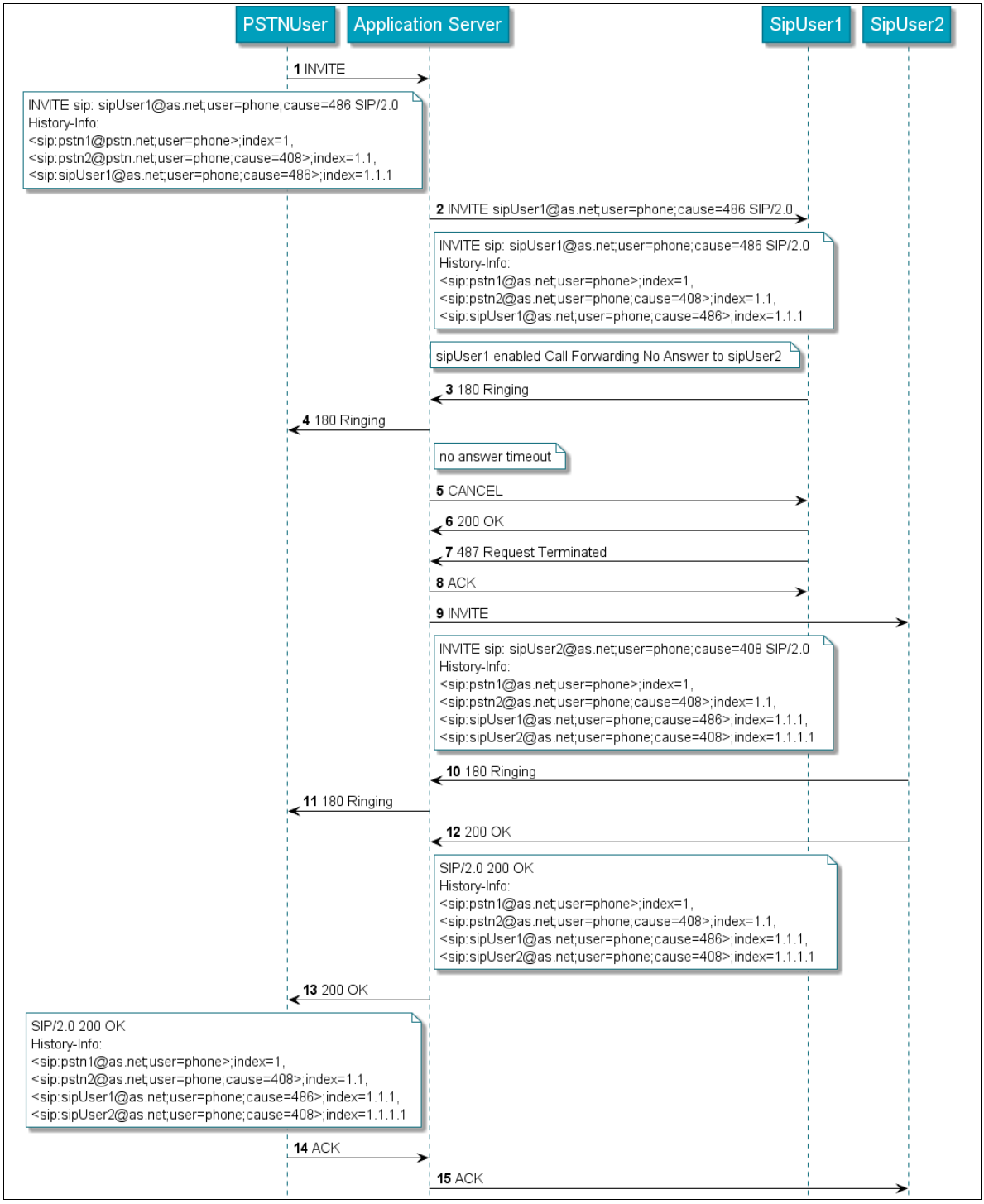

Figure 10 History-Info in 200 OK Accepted

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).

©2019 CISCO SYSTEMS, INC.

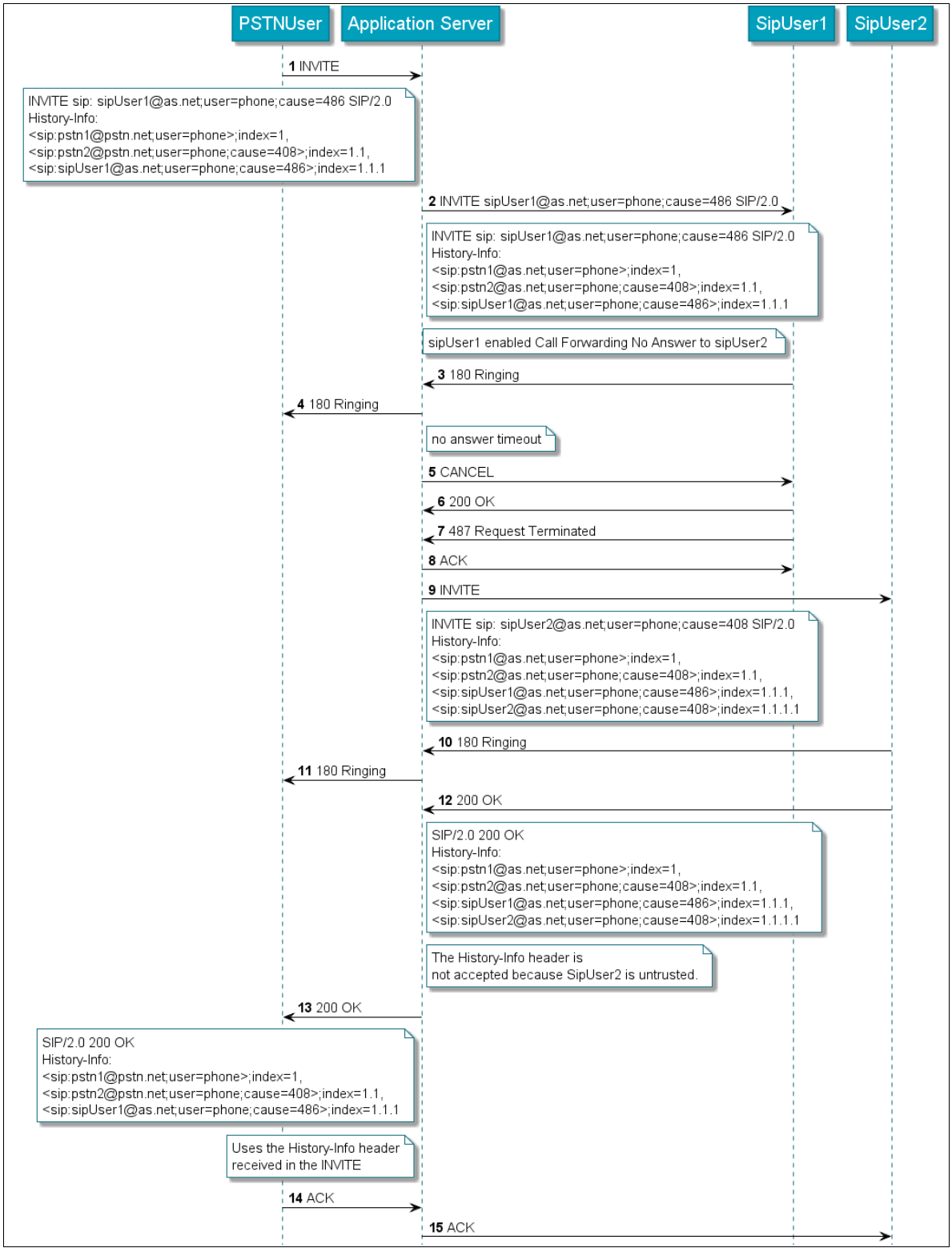

Figure 11 History-Info in 200 OK Not Accepted

©2019 CISCO SYSTEMS, INC.

3.7 Priority Alerting and Ring Splash

Cisco BroadWorks supports priority alerting/distinctive ringing to access devices. Priority alerting/distinctive ringing allows a user to provision call criteria to provide a distinctive ring or call waiting alert tone when a call is received that meets the criteria. The criteria may include information such as the calling line identification, time of day, day of week, and so on.

Cisco BroadWorks provides this service by making use of the *Alert-Info* header as defined in *RFC 3261*, section 20.4 in the INVITE request. However, instead of providing a URL for the ringing *.WAV* file to play in the network, Cisco BroadWorks provides a URL specifying the access device and a distinctive ringing indicator. For example, the *Alert-Info* header as specified in *RFC 3261* might look like the following example.

Alert-Info: <http://www.example.com/sounds/moo.wav>

An *Alert-Info* sent by Cisco BroadWorks would look like the following example.

Alert-Info: <http://127.0.0.1/Bellcore-dr2>

A device receiving a URL referencing itself via the loopback IP address, 127.0.0.1, with the Bellcore keyword should play the appropriate ringing pattern as stated in the following section.

3.7.1 Priority Ringing on Device and Ring Splash

Cisco BroadWorks supports URIs for three priority alerting cadences and one ring splash cadence as specified in the *LSSGR, GR-506-CORE, section 14*, as well as a "silent" alerting indication. Cisco BroadWorks does not send an *Alert-Info* header when the standard ringing pattern should be used. Cisco BroadWorks may send the following *Alert-Info* URIs for the various alerting signals:

- *Alert-Info*: <http://127.0.0.1/Bellcore-dr2>
- *Alert-Info*: <http:// 127.0.0.1/Bellcore-dr3>
- *Alert-Info*: <http:// 127.0.0.1/Bellcore-dr4>
- *Alert-Info*: <http:// 127.0.0.1/Bellcore-dr5>
- *Alert-Info*: <http://127.0.0.1/silent>

For analog ground-start and loop-start lines, the ringing patterns should adhere to the requirements in *GR-506-CORE*. The following table maps the URIs sent by Cisco BroadWorks to the five ringing patterns specified in *GR-506-CORE section 14*.

Cisco BroadWorks URL	Pattern ID	Pattern	Cadence	Minimum Duration (ms)	Nominal Duration (ms)	Maximum Duration (ms)
	1	Ringing Silent	2 s on 4 s off	1800 3600	2000 4000	2200 4400
Bellcore-dr2	2	Ringing Silent Ringing Silent	Long Long 	630 315 630 3475	800 400 800 4000	1025 525 1025 4400

©2019 CISCO SYSTEMS, INC.

Cisco BroadWorks URL	Pattern ID	Pattern	Cadence	Minimum Duration (ms)	Nominal Duration (ms)	Maximum Duration (ms)
Bellcore-dr3	3	Ringing Silent Ringing Silent Ringing Silent	Short Short Long 	315 145 315 145 630 2975	400 200 400 200 800 4000	525 525 525 525 1025 4400
Bellcore-dr4	4	Ringing Silent Ringing Silent Ringing Silent	Short Long Short 	200 145 800 145 200 2975	300 200 1000 200 300 4000	525 525 1100 525 525 4400
Bellcore-dr5	5	Ringing	See Note 1	450	500	550
NOTE: This ringing pattern is used for ring splash and consists of one single ringing burst.						

The URI *http://127.0.0.1/silent* is vendor-specific and is not defined in *GR-506-CORE*. It indicates that the device should start a silent alerting signal, which may consist of a visual signal.

Devices other than analog lines should attempt to adhere to the above requirements as best as they are able to, given the specific characteristics of the device.

Bellcore-dr1 is the standard cadence for North American ringing. Cisco BroadWorks does not include the *Alert-Info* header when the standard ringing cadence is requested.

Bellcore-dr5 is only sent for ring splash. This pattern consists of a single ringing burst. When a device receives an INVITE with an *Alert-Info* header with Bellcore-dr5, the device should apply the single ringing burst and wait for the originating User Agent to CANCEL the call. The CANCEL request contains the following *Reason* header to keep the missed call logs on the phone synchronized: *Reason:SIP;cause=200;text="Ring Splash"*. If the user answers the phone during this interval, the user hears silence. Note that the calling line display information for ring splash indicates that the call is for ring splash (*Figure 12*). For Call Center Hold Reminder, the ring splash INVITE request is sent only to devices that provide their own call control services, if the phone is involved in a single call that is held.

©2019 CISCO SYSTEMS, INC.

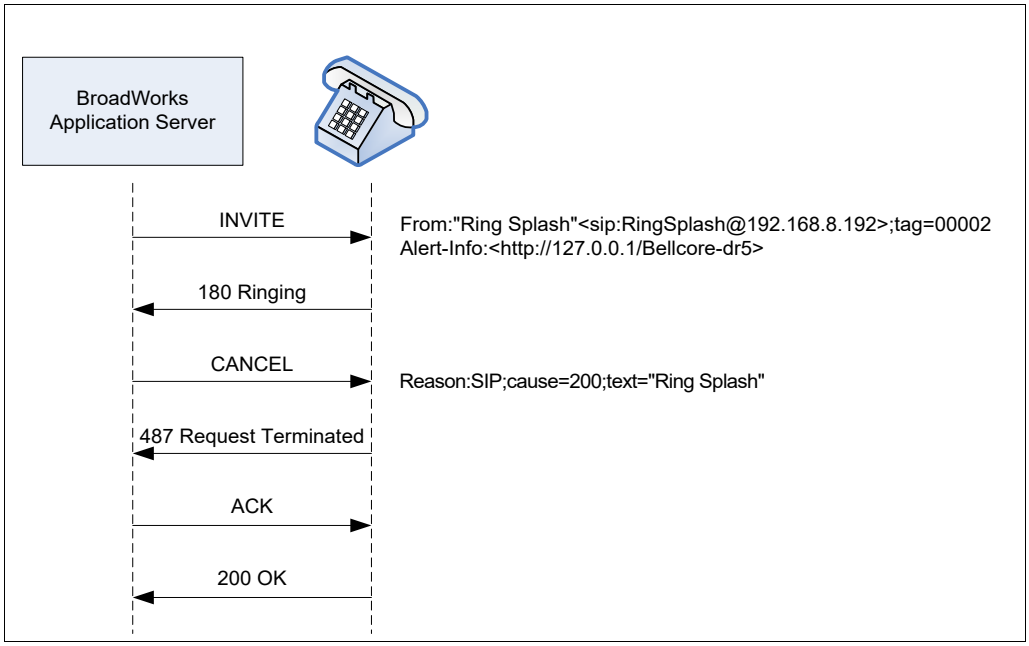

Figure 12 Ring Splash Call Flow

For devices that provide their own call control services, Cisco BroadWorks sends the *Alert-Info* header for any call that triggers distinctive ringing, independent of whether the phone is already involved in a call. These devices must support the functionality described in section *3.7.2 Priority Call Waiting Tone on Device* and map the *Alert-Info* header URL to the appropriate cadence as specified in *GR-506-CORE*.

For devices that rely on Cisco BroadWorks to provide call control services, Cisco BroadWorks only sends an INVITE request when the device is idle. Hence, the *Alert-Info* header is only included when a call triggers distinctive ringing and the device is idle. These types of devices must support the functionality described in section *3.7.2 Priority Call Waiting Tone on Device* using the INFO request for Call Waiting.

3.7.2 Priority Call Waiting Tone on Device

GR-506-CORE also specifies requirements for call waiting tones. For SIP phones and other intelligent devices which provide their own call control, Cisco BroadWorks sends *Alert-Info* headers in INVITE requests to devices for subsequent call waiting calls, with priority alerting as specified above in section *3.7.1 Priority Ringing on Device*. Upon encountering an additional call, the device should adhere to the cadences, as specified in *GR-506-CORE, section 14.*

For other devices which rely on Cisco BroadWorks call control, Cisco BroadWorks sends an INFO request with the following keyword in the message body, as specified in section *3.17 SIP INFO Method (RFC 2976)*.

- CallWaitingTone1
- CallWaitingTone2
- CallWaitingTone3
- CallWaitingTone4

©2019 CISCO SYSTEMS, INC.

For analog ground-start and loop-start lines, the call waiting tone patterns should adhere to the requirements in *GR-506-CORE*. The following table maps the URLs sent by Cisco BroadWorks to the four call waiting tone patterns specified in *GR-506-CORE section 14*. The frequency of a call-waiting tone should be 440 Hz.

Cisco BroadWorks URL	Pattern ID	Pattern	Minimum Duration (ms)	Nominal Duration (ms)	Maximum Duration (ms)
CallWaitingTone1	1	Tone On	270	300	330
Bellcore-dr2/ CallWaitingTone2	2	Tone On Tone Off Tone On	90 90 90	100 100 100	110 110 110
Bellcore-dr3/ CallWaitingTone3	3	Tone On Tone Off Tone On Tone Off Tone On	90 90 90 90 90	100 100 100 100 100	110 110 110 110 110
Bellcore-dr4/ CallWaitingTone4	4	Tone On Tone Off Tone On Tone Off Tone On	90 90 270 90 90	100 100 300 100 100	110 110 330 110 110

Devices other than analog lines should attempt to adhere to the above requirements as best as they are able to, given the specific characteristics of the device.

Bellcore-dr1 is the standard cadence for North American ringing. BroadWorks does not include the *Alert-Info* header when the standard ringing cadence is requested. However, for devices using Cisco BroadWorks call control services, Cisco BroadWorks includes the CallWaitingTone1 keyword in the INFO body for call waiting calls using the standard call waiting tone.

Cisco BroadWorks does not define a CallWaitingTone5 tone, since a ring splash should not occur when the user is on an active call (for devices that rely on Cisco BroadWorks call control).

©2019 CISCO SYSTEMS, INC.

3.8 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

3.8.1 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 *3.9 SIP Forking* provides details concerning the offer/answer model as it applies to SIP forking.
- Section *3.11 Reliability of Provisional Responses in SIP (RFC 3262)* provides details concerning the offer/answer model as it applies to reliable provisional responses.
- Section *3.12 Session Initiation Protocol (SIP) UPDATE Method (RFC 3311)* provides details concerning the offer/answer model as it applies to the SIP UPDATE method.

Although 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.

3.8.2 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]”.

©2019 CISCO SYSTEMS, INC.

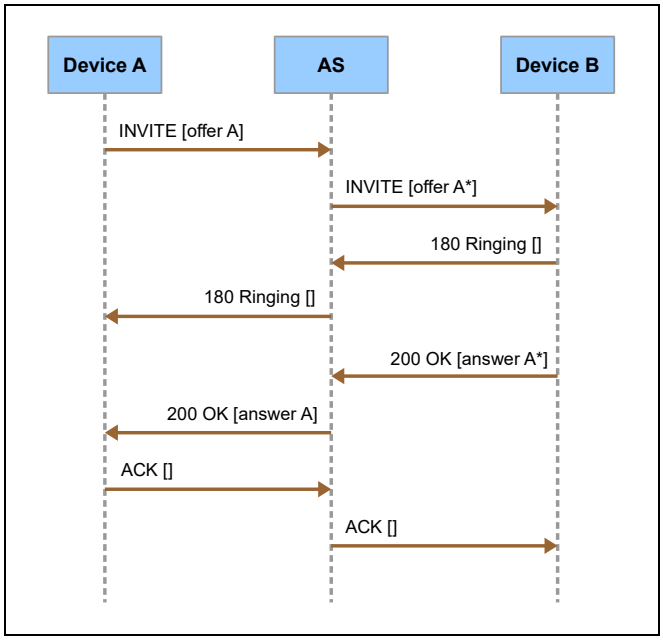

Figure 13 Basic Offer/Answer Scenario with Offer in INVITE Request

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.

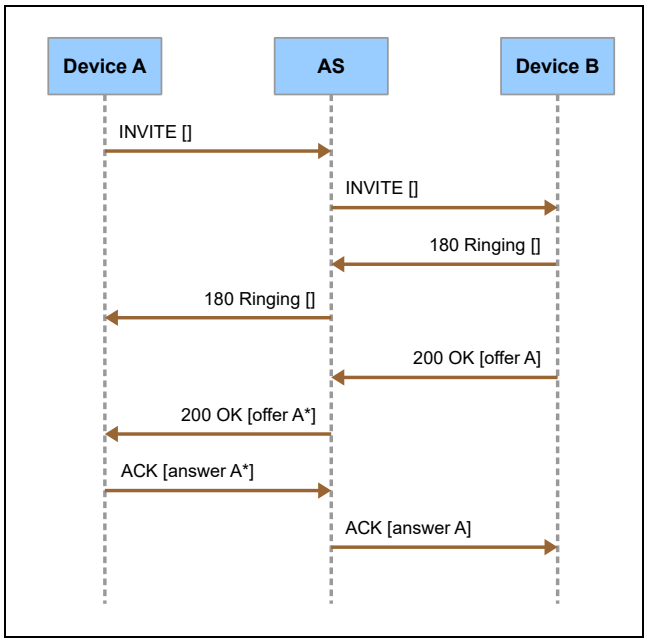

Figure 14 Basic Offer/Answer with Offer in 200 Response

©2019 CISCO SYSTEMS, INC.

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.

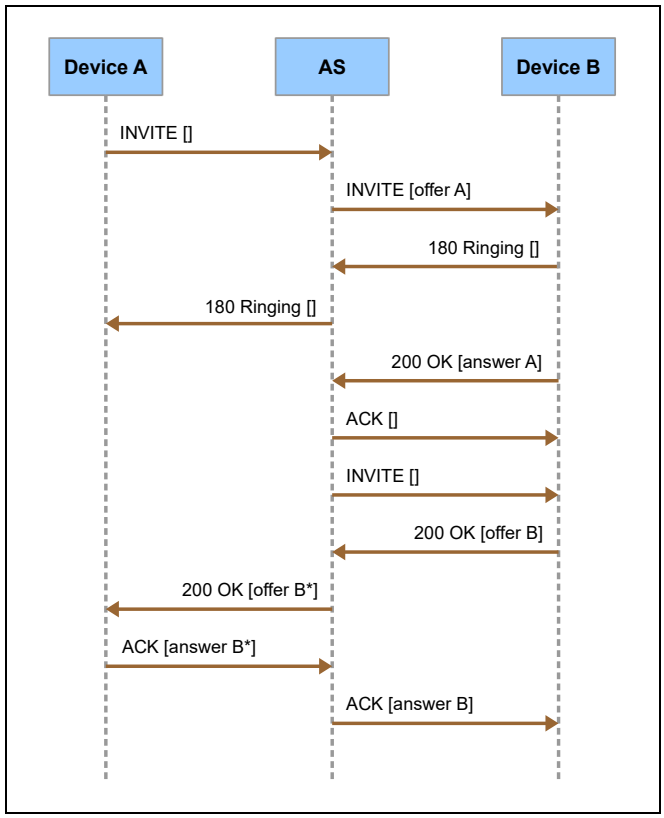

Figure 15 Basic Offer/Answer with Offer in 200 Response, Alternate Scenario

Cisco BroadWorks supports the remaining four offer/answer patterns presented in RFC 6337, including the patterns that involve PRACK (see section *3.11 Reliability of Provisional Responses in SIP (RFC 3262)*) and UPDATE (see section *3.12 Session Initiation Protocol (SIP) UPDATE Method (RFC 3311)*).

©2019 CISCO SYSTEMS, INC.

3.9 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

3.9.1 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.

3.9.2 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.

3.9.3 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 arises when a downstream proxy server sends multiple provisional responses that create early dialogs. The second situation may arise 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.

©2019 CISCO SYSTEMS, INC.

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 in the following section.
- 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 in the following section. 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.

©2019 CISCO SYSTEMS, INC.

- 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".

©2019 CISCO SYSTEMS, INC.

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 16). 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 17). 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.

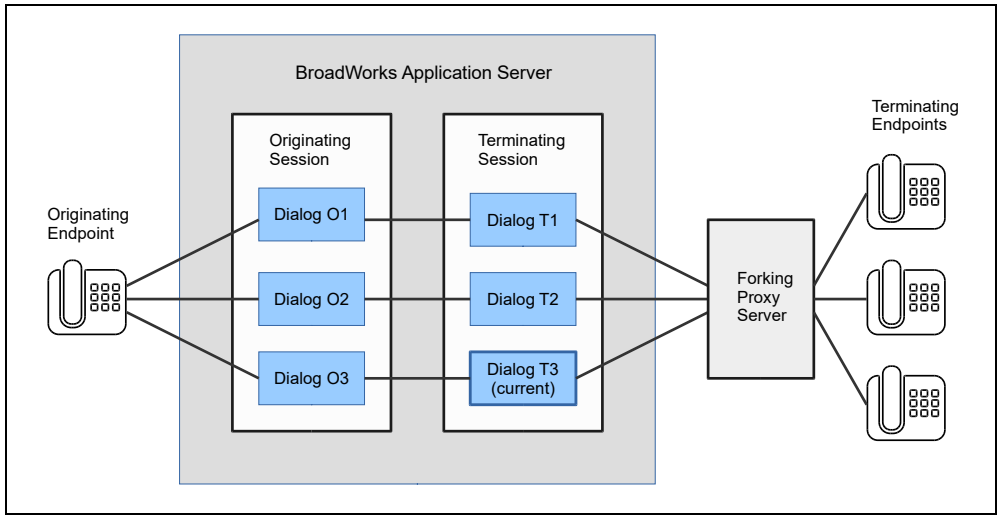

Figure 16 Originating Session with Multiple Dialog Support

©2019 CISCO SYSTEMS, INC.

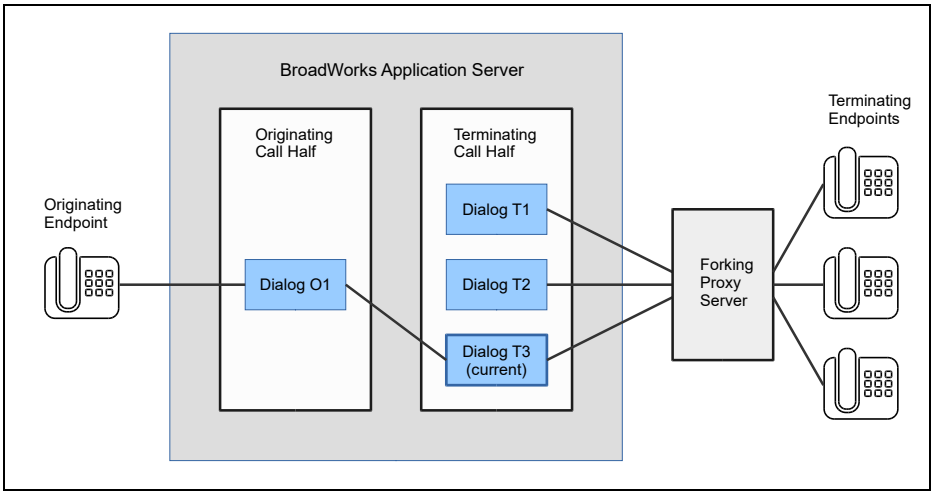

Figure 17 Originating Session with Single Dialog Support

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.

3.9.4 199 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.)

©2019 CISCO SYSTEMS, INC.

- 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 ININVTE request, even if it received *Supported* with "199" in an incoming INVITE request.
- Cisco BroadWorks ignores all 199 responses.

3.9.5 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.

©2019 CISCO SYSTEMS, INC.

More information about Cisco BroadWorks forking services is provided in section *3.36 Cisco BroadWorks P-Early-Media Header Support (RFC 5009)*, which describes forking in the context of early media.

3.9.6 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.

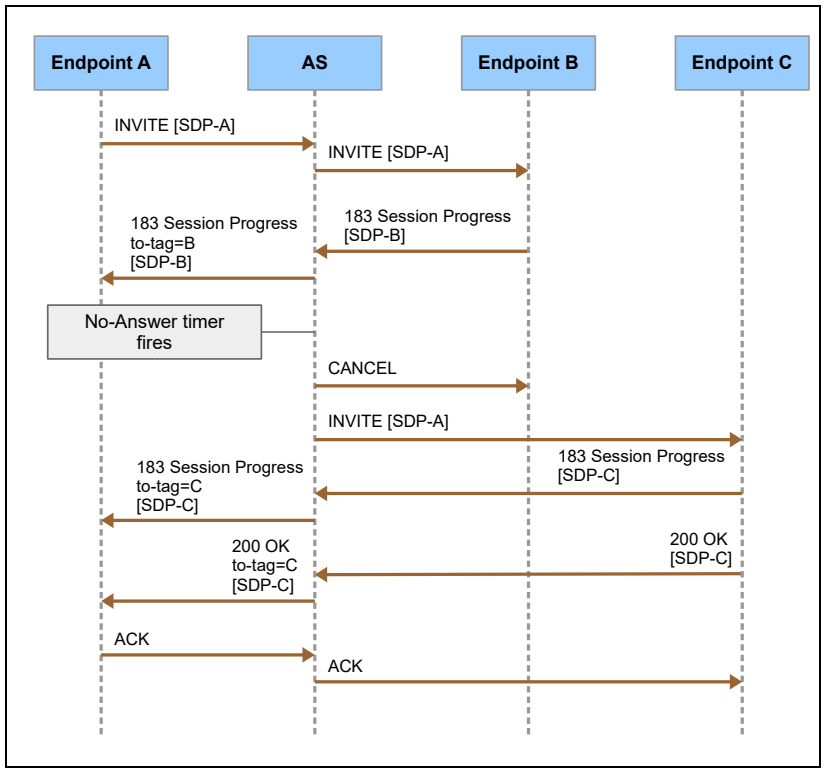

Figure 18 Call Forwarding No Answer, Multiple Dialogs

©2019 CISCO SYSTEMS, INC.

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.

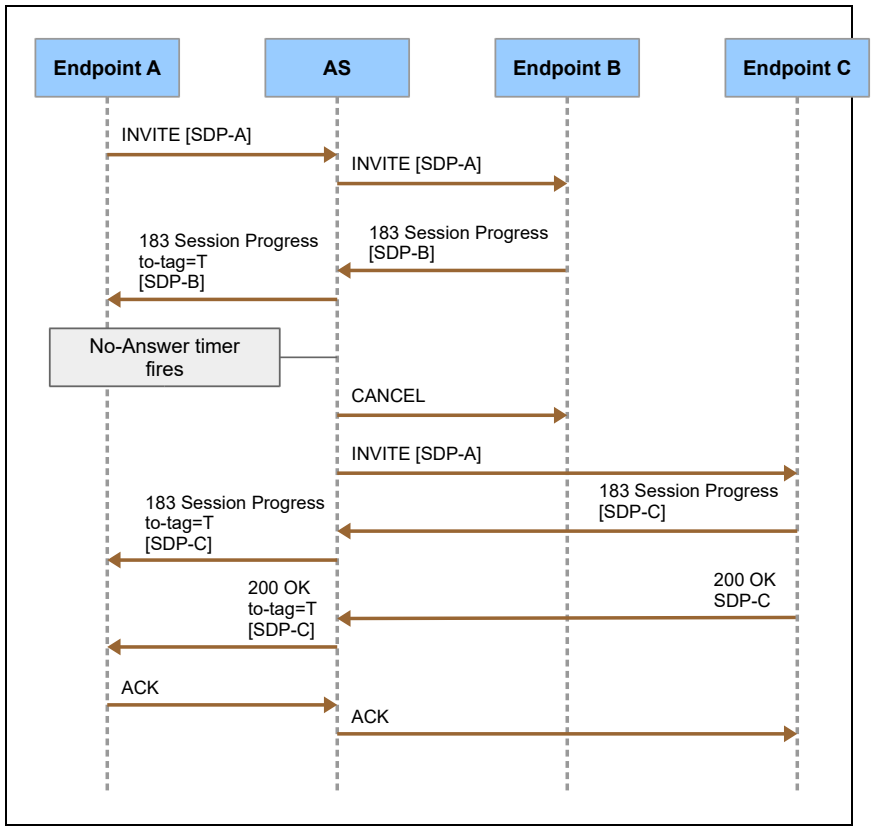

Figure 19 Call Forwarding No Answer, Single Dialog

©2019 CISCO SYSTEMS, INC.

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.

©2019 CISCO SYSTEMS, INC.

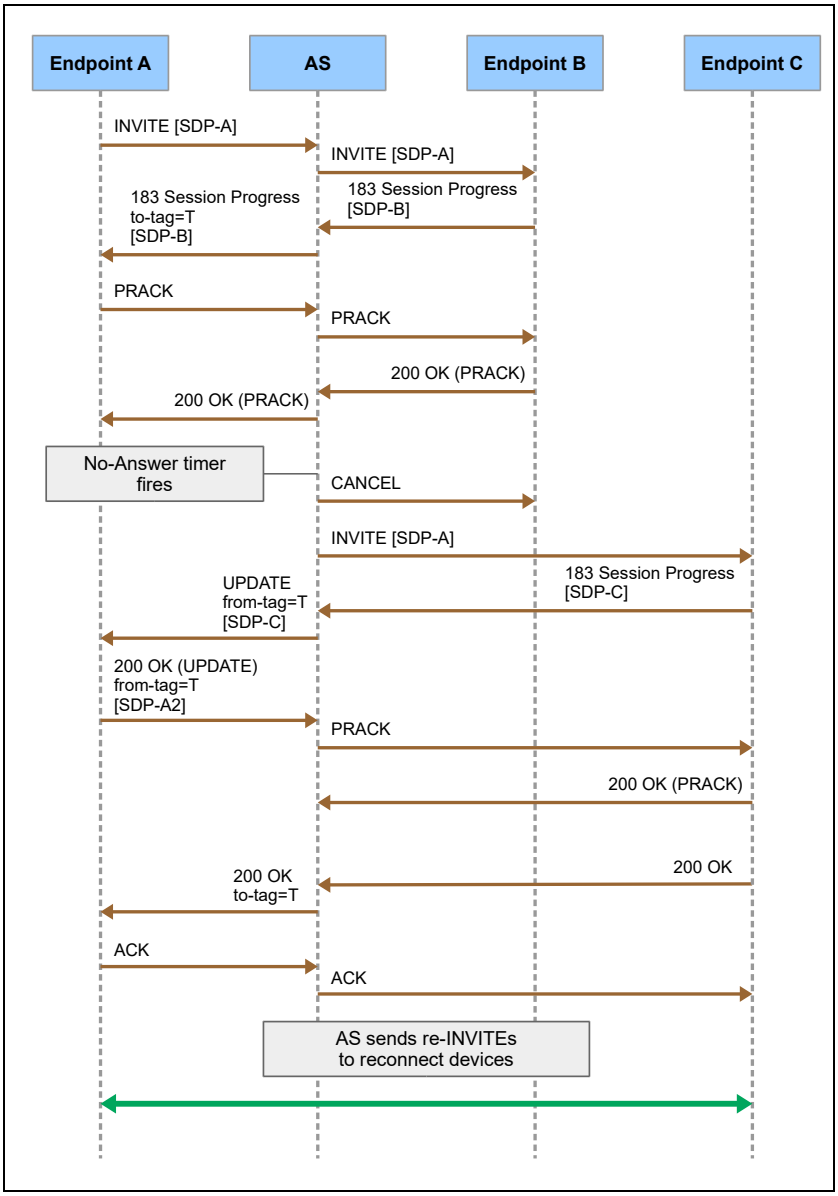

Figure 20 Call Forwarding No Answer, Single Dialog with UPDATE

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. To effect a proper offer/answer exchange, Cisco BroadWorks sends re-INVITE requests to Endpoint C and Endpoint A immediately after answer.

©2019 CISCO SYSTEMS, INC.

3.10 Early Media Transitions

Reference Documents:

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

3.10.1 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 depicted in the following call flow diagram.

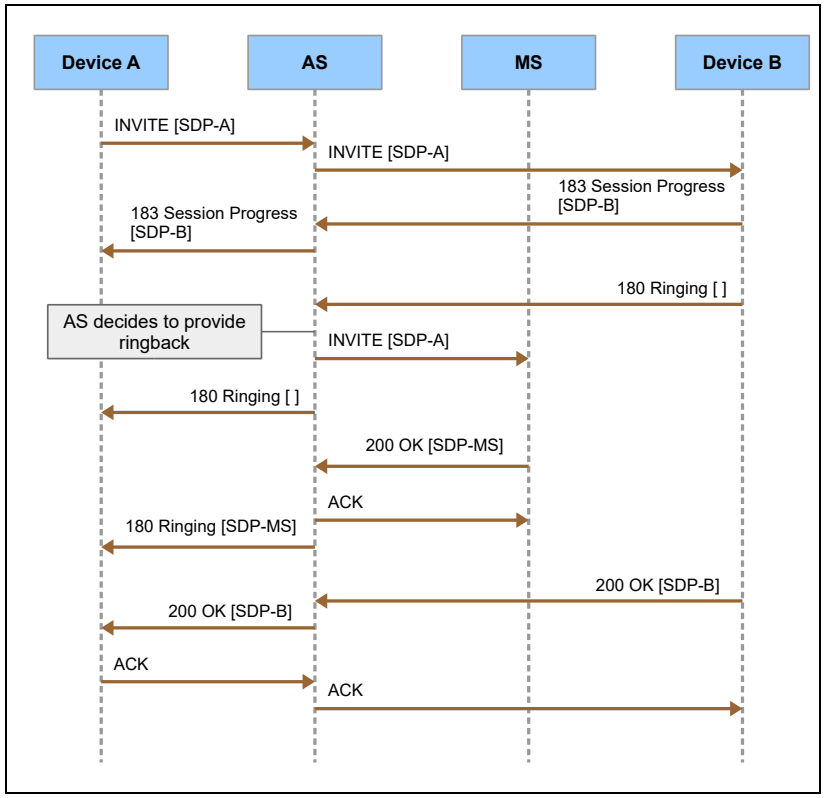

Figure 21 Early Media Transition – RFC 3398 Support Disabled

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.

©2019 CISCO SYSTEMS, INC.

- 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 section *3.9 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 depicted in the following call flow diagram.

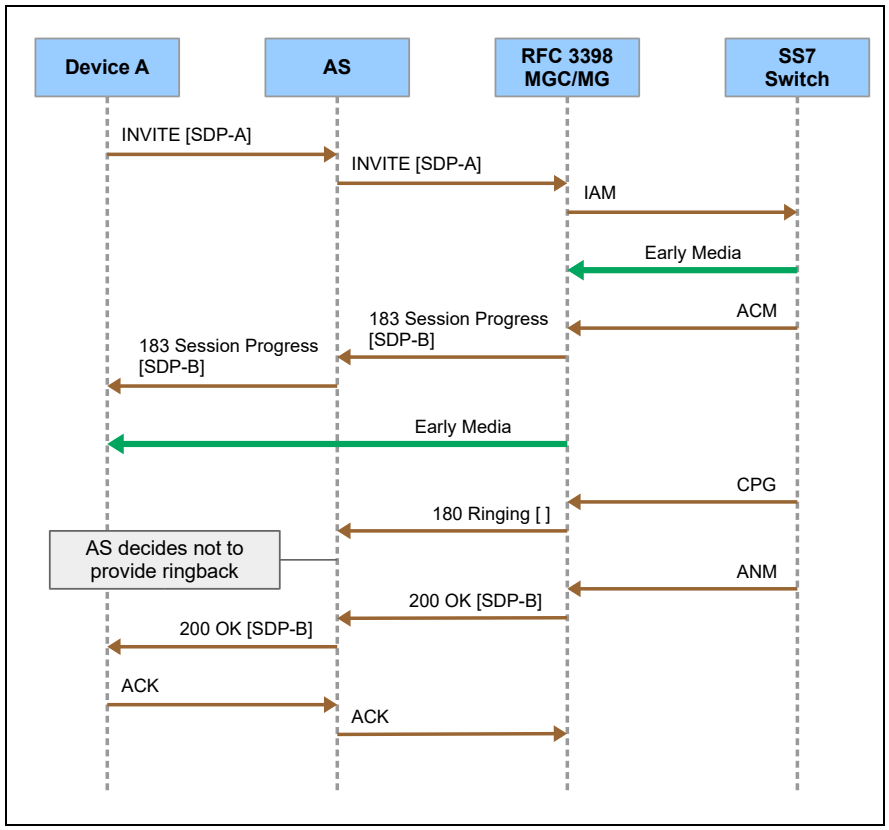

Figure 22 Early Media Transition – RFC 3398 Support Enabled

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.

©2019 CISCO SYSTEMS, INC.

3.10.2 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.

3.10.3 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.

3.10.4 Interactions with the 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 *3.36 Cisco BroadWorks P-Early-Media Header Support (RFC 5009)*.

©2019 CISCO SYSTEMS, INC.

3.11 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

3.11.1 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's set to "false". By default, *100rel* is set to "true". This section describes Cisco BroadWorks behavior when it's 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. Cisco BroadWorks supports the scenarios described in RFC 6337. These scenarios are covered in the call flows in the following list.

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".

3.11.2 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 depicts the scenario.

©2019 CISCO SYSTEMS, INC.

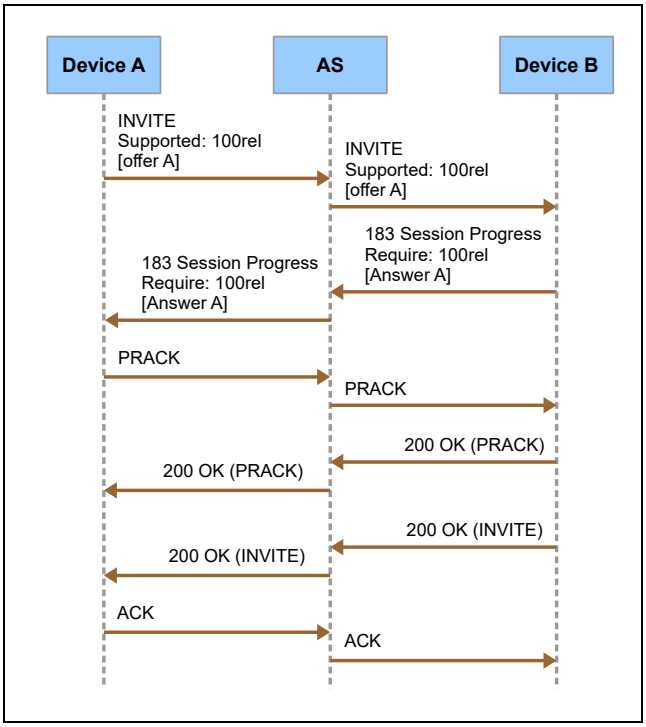

Figure 23 Offer/Answer with Answer in Reliable Provisional Response

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 depicts the scenario.

©2019 CISCO SYSTEMS, INC.

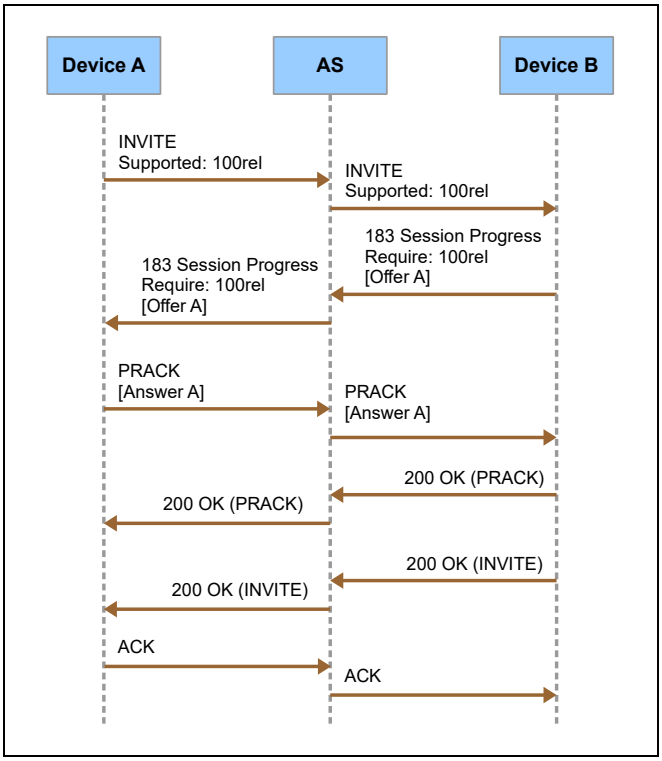

Figure 24 Offer/Answer with Offer in Reliable Provisional Response

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 depicted in the following call flow diagram.

©2019 CISCO SYSTEMS, INC.

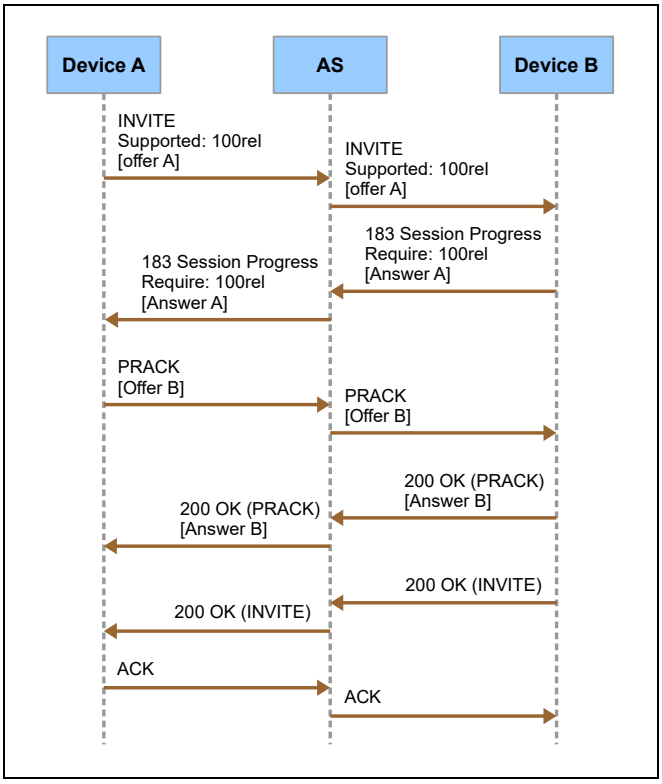

Figure 25 Offer/Answer with Second Offer in PRACK

©2019 CISCO SYSTEMS, INC.

3.12 Session Initiation Protocol (SIP) 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*.

Cisco BroadWorks may send an UPDATE request for a session-timer refresh or a Cisco BroadWorks session audit refresh, provided the endpoint supports UPDATE.

In addition, in the cases where an Access 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.

3.12.1 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

©2019 CISCO SYSTEMS, INC.

3.12.1.1 RFC 3311 - Two SIP Early Dialogs

The offer received from the Access Device in an UPDATE request is transmitted to the other SIP endpoint in an UPDATE request. The SIP endpoint provides the answer back to Cisco BroadWorks in the *200* UPDATE response. Cisco BroadWorks transmits the answer back to the Access Device in the *200* UPDATE response.

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

- 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.

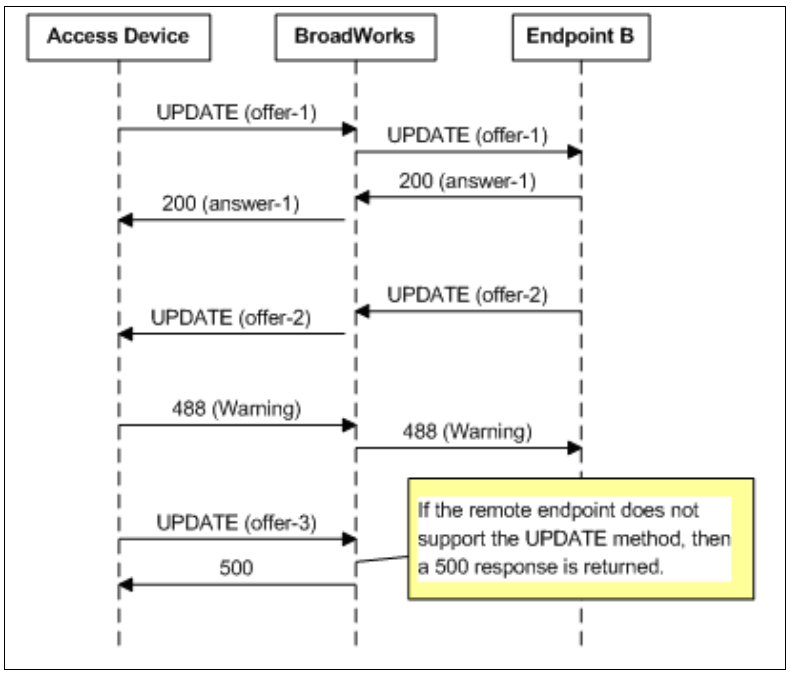

Figure 26 Two SIP Early Dialogs

3.12.1.2 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.

3.12.1.2.1 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* response.

©2019 CISCO SYSTEMS, INC.

- If the terminating endpoint rejects the offer, then Cisco BroadWorks sends a *488* response 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 Cisco BroadWorks sends a *500* response 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.

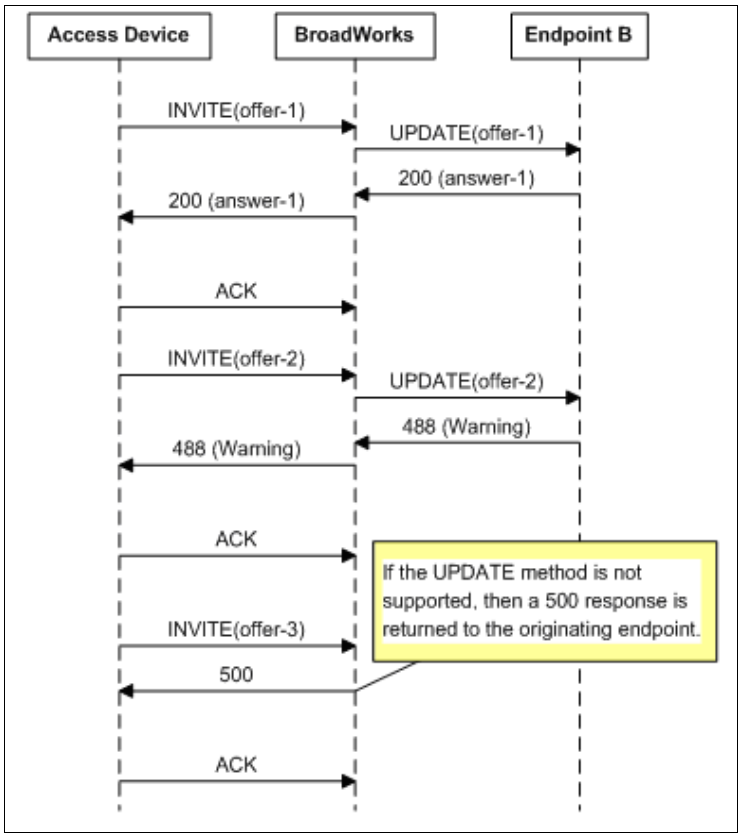

Figure 27 Offer from SIP Endpoint with Established Dialog

©2019 CISCO SYSTEMS, INC.

3.12.1.2.2 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 the support indicated in the endpoint (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. 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.

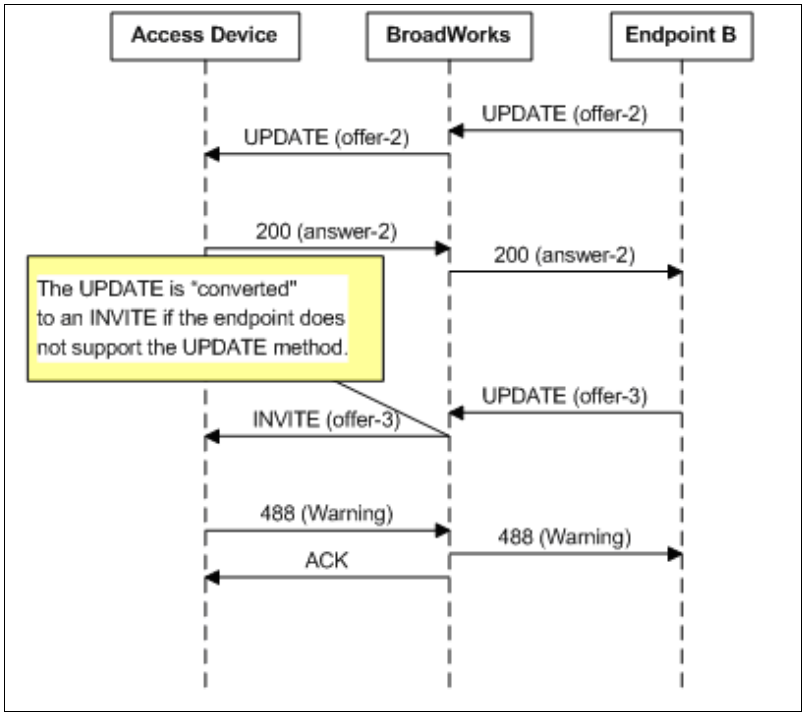

Figure 28 Offer from SIP Endpoint with Early Dialog

©2019 CISCO SYSTEMS, INC.

3.12.1.3 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 support is indicated in the endpoint (that is, *Allow* header).

- If the Access 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.

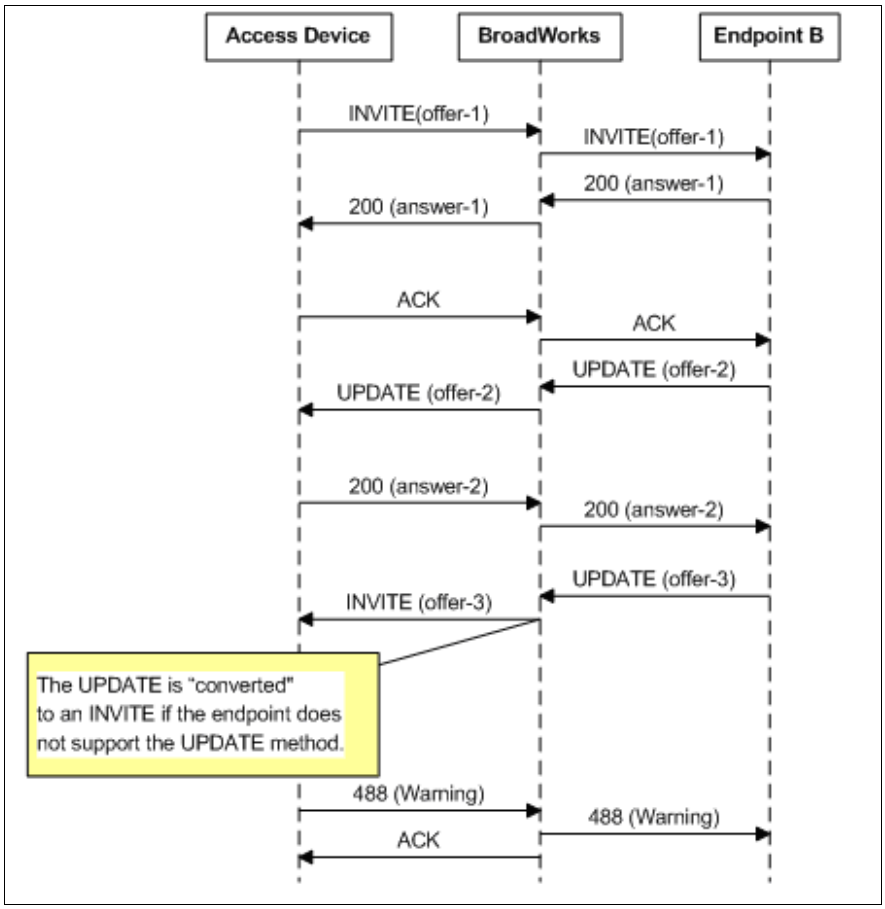

Figure 29 Two SIP Established Dialogs

©2019 CISCO SYSTEMS, INC.

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

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

RFC 3959/3960 defines 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.

3.13.1 Call Flows

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

- Early session
- Early session with forking

©2019 CISCO SYSTEMS, INC.

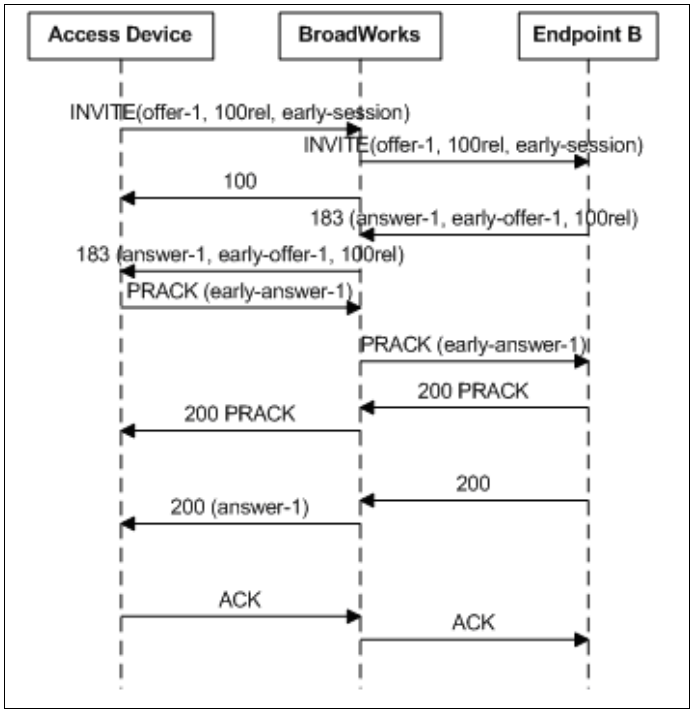

Figure 30 Early Session

©2019 CISCO SYSTEMS, INC.

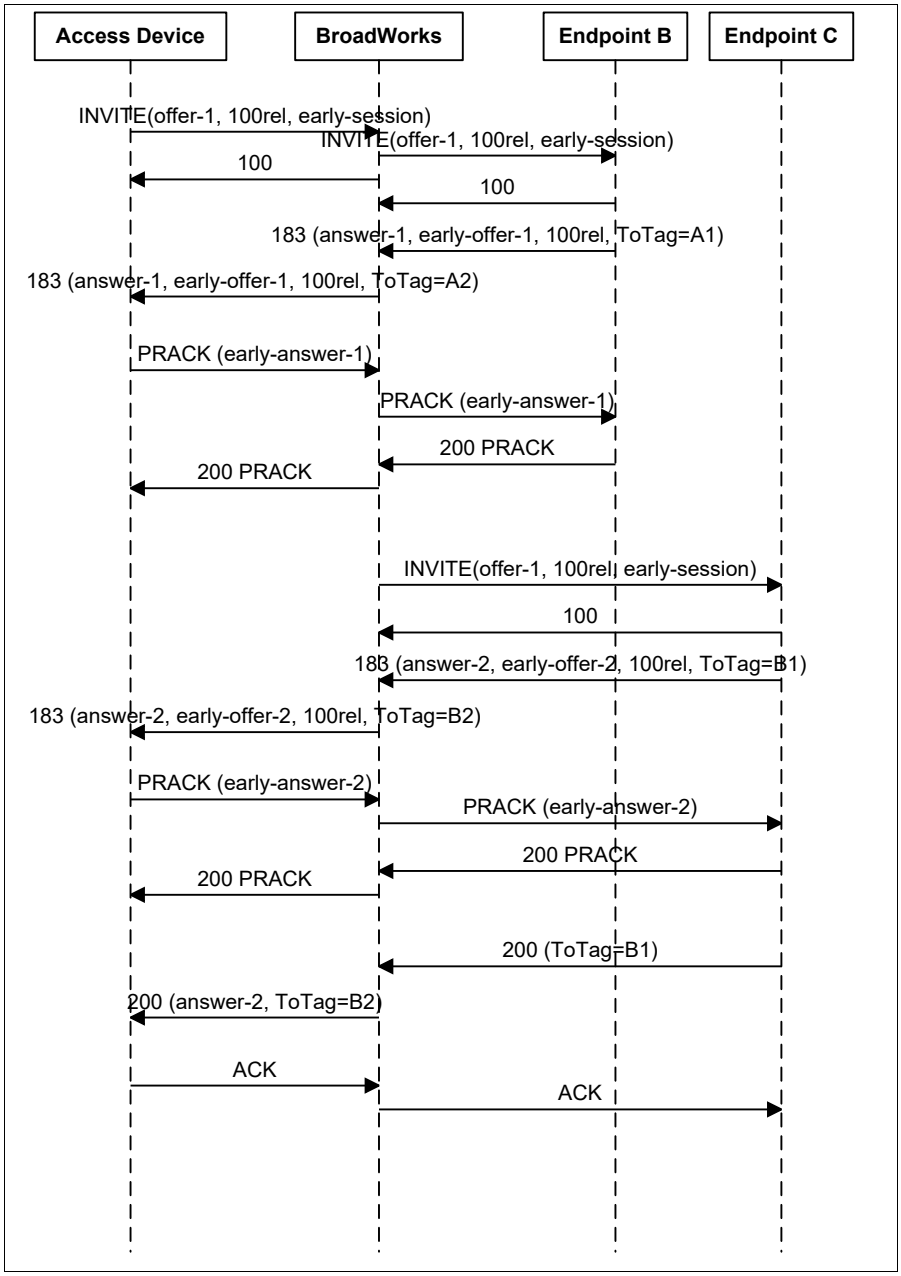

Figure 31 Early Session with Forking

©2019 CISCO SYSTEMS, INC.

3.14 Session Timers in Session Initiation Protocol (SIP) (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 this parameter is “true”, Cisco BroadWorks advertises support of the Session Timer functionality. When this parameter 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 includes the *session-expires* header. Cisco BroadWorks follows *RFC 4028* to determine the refresher for the session; however, the remote user agent is always 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 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”.

©2019 CISCO SYSTEMS, INC.

3.15 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 *3.1.3.1 Differences between UDP and TCP Transports for SIP*.

3.15.1 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.

©2019 CISCO SYSTEMS, INC.

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

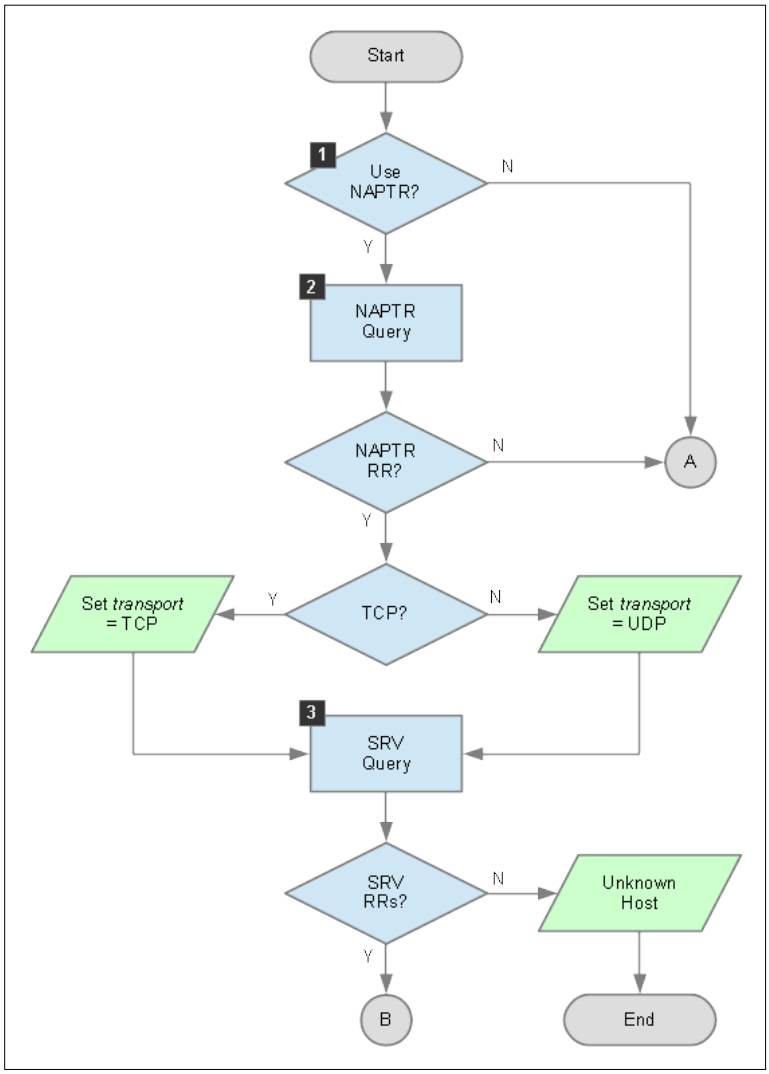

Figure 32 Flow Diagram for DNS NAPTR Query

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).

©2019 CISCO SYSTEMS, INC.

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)[4].

   – 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.

[4] 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.

©2019 CISCO SYSTEMS, INC.

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.

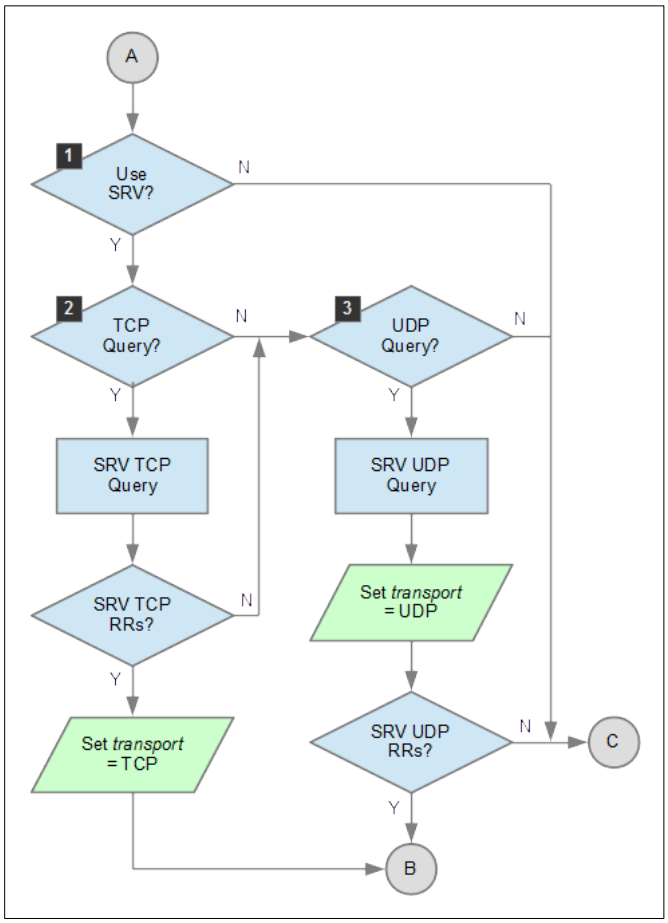

Figure 33 Flow Diagram for DNS SRV Query

The following notes explain the callouts in the 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.

©2019 CISCO SYSTEMS, INC.

- 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 figure.

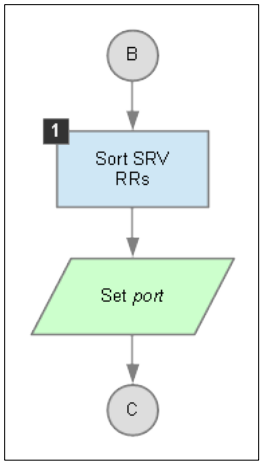

Figure 34 Flow Diagram for DNS SRV Record Processing

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 figure before performing an A query or AAAA query.

©2019 CISCO SYSTEMS, INC.

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.

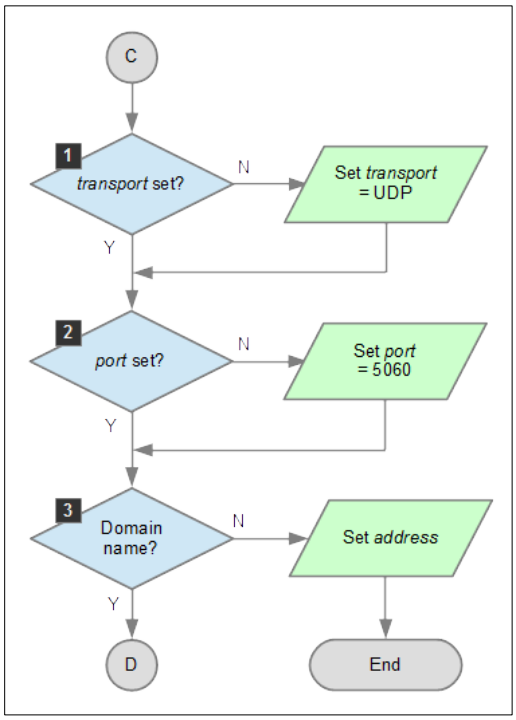

Figure 35 Flow Diagram for DNS A/AAAA Query Preparation

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 might be an IPv4 address or an IPv6 address, in which case Cisco BroadWorks uses the explicit address.

©2019 CISCO SYSTEMS, INC.

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

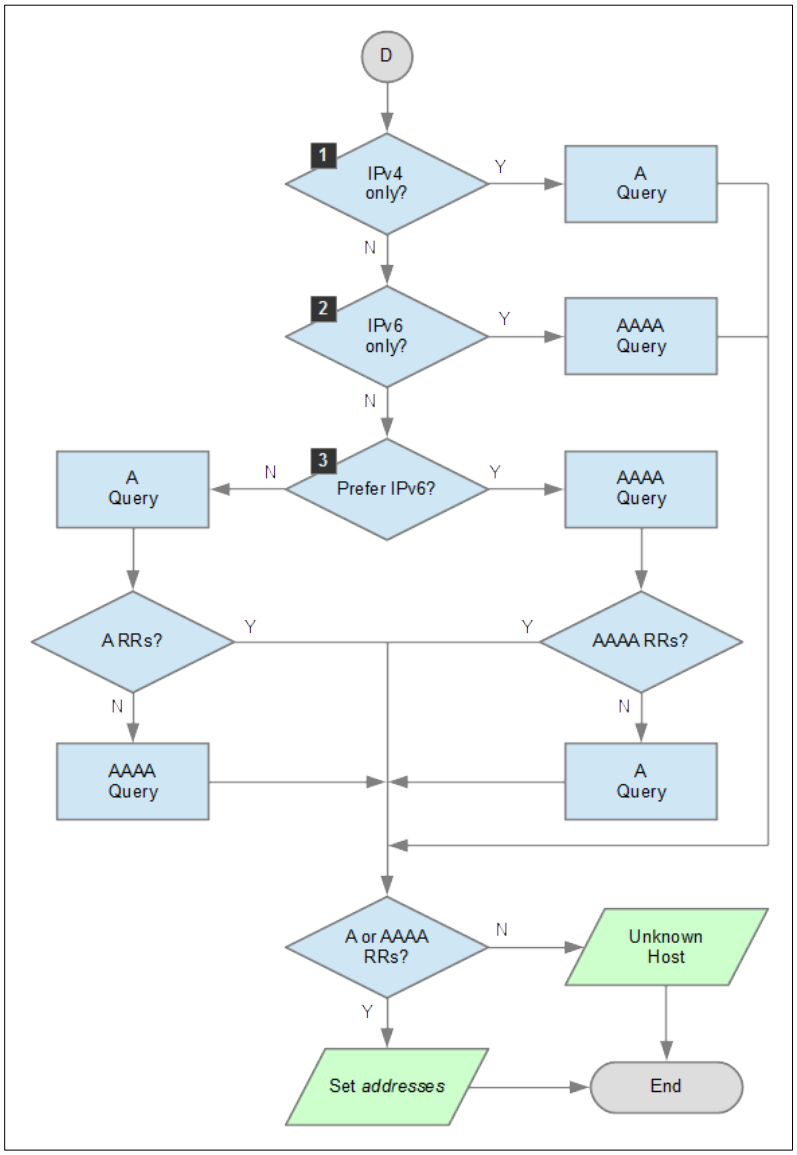

Figure 36 Flow Diagram for DNS A/AAAA Query

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 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”)

©2019 CISCO SYSTEMS, INC.

- *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 ***
```


©2019 CISCO SYSTEMS, INC.

3.16 Best Current Practices for Third Party Call Control (3PCC) in Session Initiation Protocol (SIP) (RFC 3725)

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

©2019 CISCO SYSTEMS, INC.

3.17 SIP INFO Method (RFC 2976)

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. If the applications use SIP signaling, such information is conveyed in SIP INFO requests.

For more details on message body proxy capabilities, see section *3.28 Cisco BroadWorks Media Type Support*.

In addition to proxying INFO requests, Cisco BroadWorks processes some specific INFO content. Cisco BroadWorks uses the INFO method 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 methods 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 install and upgrades.
Application/broadsoft	Interpreted and never proxied.
others	Proxied if configured under *AS_CLI/Interface/SIP/ContentType*.

Table 1 Special INFO Messages

3.17.1 Flash-based Service Support via INFO Method

Cisco BroadWorks uses a proprietary extension to the INFO method to support flash-based user services. A device must support this extension for Cisco BroadWorks to provide flash-based services such as call waiting, call transfer, three-way calling, and so on. Specifically, the extension includes the definition of a new value for the *Content-Type* header. The new value is "application/broadsoft".

The "application/broadsoft" Content-Type allows an endpoint to notify Cisco BroadWorks that a flash hook has occurred or to direct an endpoint to play a tone, as specified by the Cisco BroadWorks Application Server.

The Content-Type of "application/broadsoft" indicates that a proprietary body is in the message. The body must be in one of the following formats for Cisco BroadWorks or the endpoint to interpret the intention: (These fields are not case sensitive.)

- event <event name>
- play tone <tone name>
- stop <tone name>

©2019 CISCO SYSTEMS, INC.

Optionally, the play tone body may contain the following body parts to communicate call waiting calling party identification information. Note that the INFO body is case insensitive.

- Calling-Name:<calling-name> where <calling-name> is a string representing the calling party's name
- Calling-Number:<calling-number> where <calling-number> is a string representing the calling party's number

The Calling-Name and Calling-Number are always included in the INFO for call waiting as long as the calling party information is available. When the information is not available, the device must populate the calling line identification signal to the analog line with the appropriate unavailable signal. When the calling party information is not available, the *Calling-Name* and/or *Calling-Number* fields are not included in the INFO method body. It is possible that the calling number may be available without the calling name and vice versa. When these conditions occur, only the information that is available is included in the INFO method body (that is, it is possible to have a *Calling-Number* field in the INFO method body without a *Calling-Name* field and vice versa).

When any portion of the calling party information is restricted, the *Calling-Name* and *Calling-Number* fields are included in the INFO method body header and are populated with "Private".

NOTE: Restricted calling party information overrides unavailable calling party information.

When the calling number is restricted and the calling name is unavailable or vice versa, both the *Calling-Name* and *Calling-Number* fields are included in the INFO method body and are populated with "Private".

```
INFO sip:2403649314@10.10.14.200:5060 SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-
10.10.14.200V5060-0-1000815249 -1646372935-985549141285
From:<sip:2403645138@applicationserver.broadworks.com;user=phone>;tag=156
9407939-1023204223358
To:"richard ricardo"<sip:2403649314@10.10.14.200;user=phone>;tag
=0003e3630c94001d20f605e8
Call-
ID:BW112343035704060202316497252421 72@applicationserver.broadworks.com
CSeq: 1000815249 INFO
Content-Length:28
Content-Type:application/broadsoft

play tone CallWaitingTone1
Calling-Name:"Private"
Calling-Number:Private
```

The *Calling-Number* field is sent in the INFO method body with the national format when the calling number and called number are within the same country code.

```
Calling-Number:2403645137
```

When the calling number and called number are in different country codes, the *Calling-Number* field is populated with the E.164 number of the calling party.

```
Calling-Number:+12403645137
```


©2019 CISCO SYSTEMS, INC.

The only event currently defined is flash hook (this is not case sensitive).

The only tones currently defined are:

- CallWaitingTone1
- CallWaitingTone2
- CallWaitingTone3
- CallWaitingTone4

NOTE: The tones do not actually specify any detail about the tone such as frequency levels, cadence, and duration. The tones are merely identifiers that should uniquely identify a physical tone that can be applied to an endpoint. The physical characteristics of the tones are defined in *GR-506-CORE, section 14*. For more information on the tone characteristics, see section *3.7.2 Priority Call Waiting Tone on Device*.

The only parameter allowed for the stop body is *CallWaitingTone*.

- A body with stop *CallWaitingTone* indicates that the access device should cease applying the call waiting tone regardless of which type of call waiting tone is applied.

Following is an example of INFO with flash body. Cisco BroadWorks never sends this type of INFO. The access device should send this INFO when a flash is detected on the analog line.

```
INFO sip:10.10.180.73:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.14.200:5060
From:"richard ricardo"<sip:2403649314@10.10.14.200;user=phone>;
tag=0003e3630c94001d20f605e8
To:<sip:2403645138@applicationserver.broadworks.com;user=phone>;tag=15694
07939-1023204223358
Call-
ID:BW112343035704060202316497252421 72@applicationserver.broadworks.com
CSeq: 1000815249 INFO
Content-Length:17
Content-Type:application/broadsoft

event flashhook
```

Following is an example of INFO with play tone body. Cisco BroadWorks sends this type of INFO when a Cisco BroadWorks subscriber has the Flash Call Waiting service assigned and a second call arrives for the Cisco BroadWorks subscriber while the subscriber is in an active call. The access device should never send this type of INFO.

```
INFO sip:2403649314@10.10.14.200:5060 SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-
10.10.14.200V5060-0-1000815249 -1646372935-985549141285
From:<sip:2403645138@applicationserver.broadworks.com;user=phone>;tag=156
9407939-1023204223358
To:"richard ricardo"<sip:2403649314@10.10.14.200;user=phone>;tag
=0003e3630c94001d20f605e8
Call-
ID:BW112343035704060202316497252421 72@applicationserver.broadworks.com
CSeq: 1000815249 INFO
Content-Length:28
Content-Type:application/broadsoft
```


©2019 CISCO SYSTEMS, INC.

```
play tone CallWaitingTone1
Calling-Name:"Fred Mertz"
Calling-Number:3019779440
```

Following is an example of INFO with stop body. Cisco BroadWorks sends this type of INFO under the following conditions:

- Cisco BroadWorks subscriber has Flash Call Waiting service assigned.
- The second call arrives for the Cisco BroadWorks subscriber while the subscriber is in an active call.
- INFO with a play tone body has been sent to the device to apply the appropriate call waiting tone.
- One of the following events occurs:
  - The calling party hangs up prior to the Cisco BroadWorks subscriber answering the waiting call.
  - The device signals a flash hook to Cisco BroadWorks, which indicates that Cisco BroadWorks should answer the incoming call and connect the device to the new incoming call.
  - A client device, such as the CommPilot Call Manager or an Xsi client, sends an answer/talk request, which indicates that Cisco BroadWorks should answer the incoming call and connect the device to the new incoming call.

NOTE: The access device should send never send this type of INFO.

```
INFO sip:2403649314@10.10.14.200:5060 SIP/2.0
Via:SIP/2.0/UDP 10.10.180.73:5060;branch=z9hG4bKBroadWorks.-1su2iau-
10.10.14.200V5060-0-1000815249 -1646372935-985549141285
From:<sip:2403645138@applicationserver.broadworks.com;user=phone>;tag=156
9407939-1023204223358
To:"richard ricardo"<sip:240364931410.10.14.200;user=phone>;tag
=0003e3630c94001d20f605e8
Call-
ID:BW112343035704060202316497252421720applicationserver.broadworks.com
CSeq: 1000815249 INVITE
Content-Length:28
Content-Type:application/broadsoft

stop CallWaitingTone
```

3.17.2 Video Support via INFO Method

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 proxies 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:

©2019 CISCO SYSTEMS, INC.

- 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. *RFC 5168* describes the encoding of this media control information in an XML payload with MIME type application/media_control+xml. For more information, see *RFC 5168 XML Schema for Media Control* [37].

The following figure shows one possible scenario where 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.

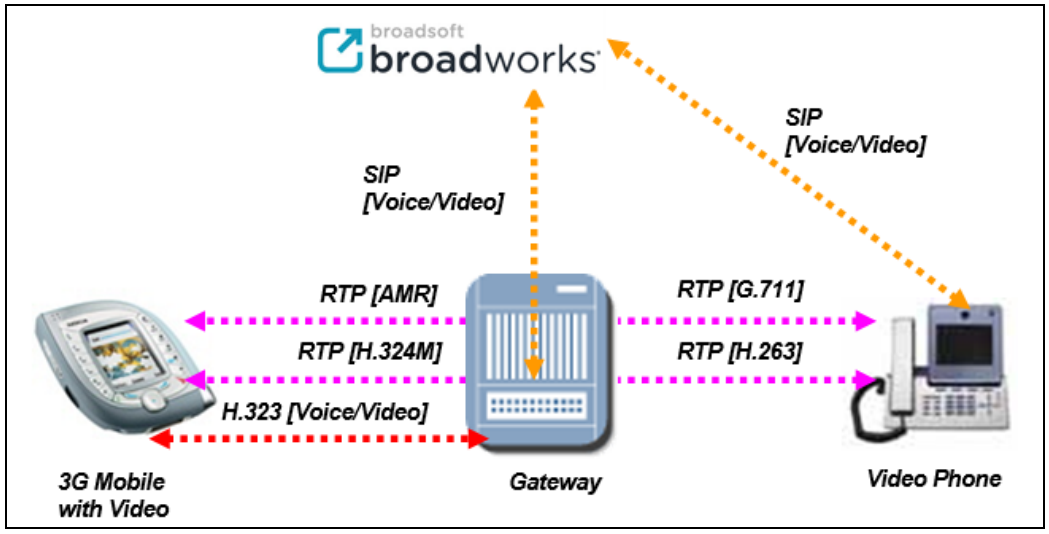

Figure 37 Cisco BroadWorks Support of INFO Proxy for Media Control with Video Applications

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 will not send the INFO request that device.

The Application Server does not proxy the INFO request if it contains a multipart body that contains an application/media_control+xml body part.

The Media Server does not support receiving INFO requests for proxying of media control for video applications. The Media Server is not a video encoder or decoder – it is a video streaming server. For that reason, the Media Server cannot act on video control messages, and therefore, ignores them. A “SIP 501 Not Implemented” response is returned for INFO requests sent to the Media Server. The Media Server will not send SIP INFO requests with video control messages.

©2019 CISCO SYSTEMS, INC.

3.17.3 DTMF Support via the INFO Method

Cisco BroadWorks supports out-of-band DTMF through the INFO method. The following Content-Type carrying DTMFs 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, the 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 repeatedly. It is imperative that a device sending a DTMF event using an INFO request does not send the same DTMF event over the RTP media stream (not modulated in the voice path or through *RFC 4733*).

Figure 38 shows a situation in which a user accesses his or her voice portal. The Application Server is configured to proxy the application/dtmf-relay to the Media Server.

©2019 CISCO SYSTEMS, INC.

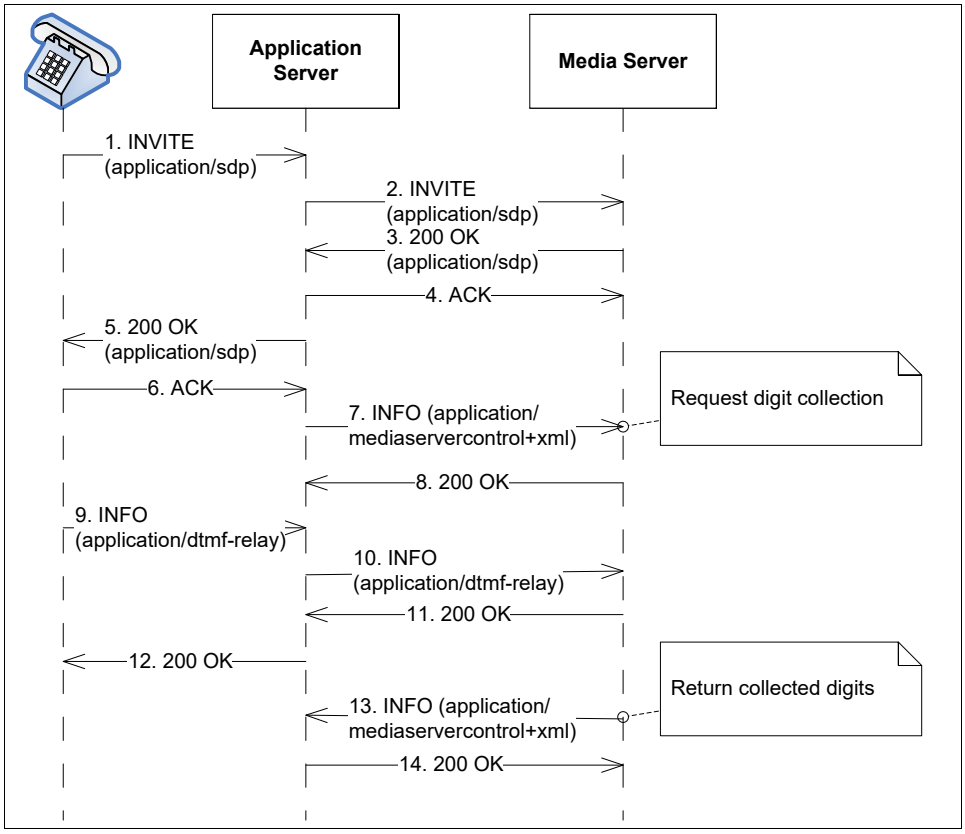

Figure 38 Media Server Processes Application/dtmf-relay

©2019 CISCO SYSTEMS, INC.

3.18 Session Initiation Protocol (SIP) Refer Method (RFC 3515)/SIP "Replaces" Header (RFC 3891)/SIP Referred-BY Mechanism (RFC 3892)

Cisco BroadWorks supports this functionality.

Cisco BroadWorks never sends a REFER request or any request with a *Replaces* header. However, Cisco BroadWorks accepts receiving a REFER request for blind transfer and a REFER request with *Replaces* header encapsulated into *Refer-to* header for transfer with consultation. Cisco BroadWorks does not support receiving a REFER request outside of a dialog.

In general, Cisco BroadWorks does not support receiving a *Replaces* header in an INVITE. Cisco BroadWorks ignores the *Replaces* header field and processes the INVITE as if the *Replaces* header is not present, regardless of the presence of the *Require* header. However, Cisco BroadWorks supports INVITE requests with a *Replaces* header in some Shared Call Appearance scenarios. For information on the use of the *Replaces* header for Shared Call Appearance, see the *Cisco BroadWorks SIP Access Side Extensions Interface Specification* [43].

Since Cisco BroadWorks is a back-to-back user agent (B2BUA), Cisco BroadWorks has knowledge of all calls to/from a particular device. As such, Cisco BroadWorks is able to accept a REFER method and perform the appropriate call control requests within Cisco BroadWorks, rather than proxy the request on to the other party in the call. This is vital for interworking SIP devices with devices, which support other protocols such as Media Gateway Control Protocol (MGCP). This is also beneficial for interworking with SIP devices which do not support the REFER method.

Access devices may use this method and header to trigger transfer services within Cisco BroadWorks. Cisco BroadWorks transparently processes the request such that the access device is unaware of any interworking required to complete the request.

©2019 CISCO SYSTEMS, INC.

Depending on configuration, Cisco BroadWorks can send or suppress the NOTIFY for the implicit subscription created by the REFER method on the dialog. When configured to do so, the Application Server sends a NOTIFY request per *RFC 3515* to suppress the implicit REFER subscription. The NOTIFY terminates the implicit subscription and contains *message/sipfrag* content as described in *RFC 3515.* When configured not to send the NOTIFY, the BYE implicitly terminates the subscription.

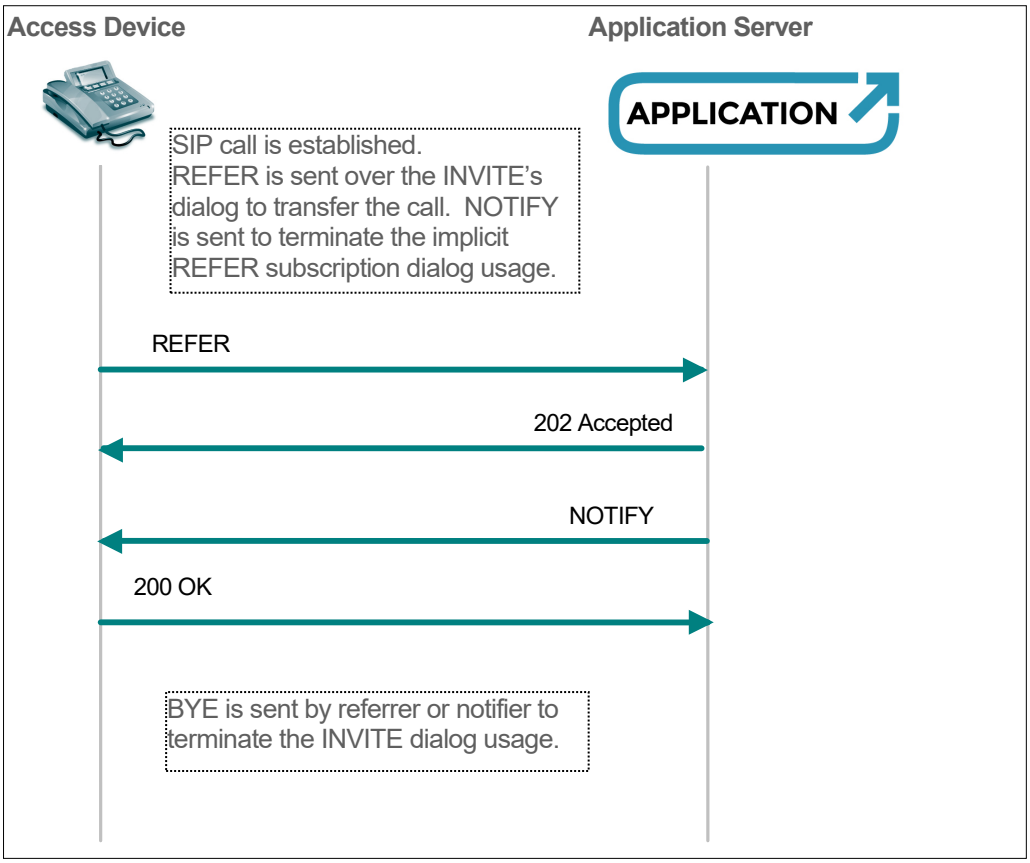

Figure 39 REFER's Implicit Subscription is Terminated with NOTIFY

The following is an example of a NOTIFY sent to terminate the implicit REFER subscription.

```
NOTIFY sip:line-port-1@192.168.40.11 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:3015550000@as.broadsoft.com;user=phone>;
tag=946677141-1240957044390-
To:<sip:line-port-1@broadsoft.com>;tag=12345888
Call-ID:BW131909796040500-565063280@as.broadsoft.com
CSeq:929489740 NOTIFY
Contact:<sip:as-cluster1.broadsoft.com>
Event:refer;id=4
Subscription-State:terminated;reason=noresource
Max-Forwards:10
Content-Type:message/sipfrag
Content-Length:20

SIP/2.0 100 Trying
```


©2019 CISCO SYSTEMS, INC.

3.19 Session Initiation Protocol (SIP) Call Control Conferencing for User Agents (RFC 4579)/Framework for Conferencing with SIP (RFC 4353)

3.19.1 SIP Conferencing in Cisco BroadWorks

Cisco BroadWorks supports conferencing both in the form of an ad hoc conference and a planned conference[5]. An ad hoc conference is a spontaneously created conference that uses a transient conference bridge. Cisco BroadWorks creates an ad hoc conference[6] when the user invokes the Three-Way Call service or the N-Way Call service, which supports up to fifteen participants. In contrast, a planned conference uses a provisioned conference bridge. Cisco BroadWorks supports planned conferences through the Meet-Me Conferencing service. In both conference types, the Cisco BroadWorks Application Server performs the role of a conference focus, while the Cisco BroadWorks Media Server performs the role of the conference mixer. For both conference types, Cisco BroadWorks allows participation by conference-unaware participants. However, for an ad hoc conference, Cisco BroadWorks also supports many of the scenarios described in *RFC 4579* for a conference-aware participant. In such scenarios, the conference-aware participant must be the participant that created the conference.

The following table covers each scenario in *RFC 4579*, describing the how scenario relates to conferencing in Cisco BroadWorks.

RFC 4579 Section	Scenario Title	Cisco BroadWorks Support
5.1	INVITE: Joining a Conference Using the Conference URI - Dial-In	Cisco BroadWorks supports this dial-in scenario for Meet-Me conferences, but not ad hoc conferences. When Cisco BroadWorks sends a response to the INVITE request, it does not add the *isfocus* parameter to the *Contact* header nor the *Allow-Events* header with the value "conference".
5.2	INVITE: Adding a Participant by the Focus - Dial-Out	Cisco BroadWorks supports this dial-out scenario for Meet-Me conferences, but not ad hoc conferences. When Cisco BroadWorks sends the INVITE request, it does not add the *isfocus* parameter to the *Contact* header or the "conference" event package name to the *Allow-Events* header.
5.3	INVITE: Manually Creating a Conference by Dialing in to a Conferencing Application	Cisco BroadWorks supports this conference creation scenario for Meet-Me conferences. For a Meet-Me conference, Cisco BroadWorks sends neither the *isfocus* parameter nor the "conference" event package name in the *Allow-Events* header.
5.4	INVITE: Creating a Conference Using Ad-Hoc SIP Methods	Cisco BroadWorks supports this scenario for ad hoc conferences. For the ad hoc conference, Cisco BroadWorks sends a 200 response to the INVITE, which has an *isfocus* parameter in the *Contact* header as well as an *Allow-Events* header with the "conference" event package name. Cisco BroadWorks does not send a 302 response to the creator's user agent, as in *RFC 4579*. See the call flows in section *3.19.3.1 Adding a Participant: REFER Request to the User* and section *3.19.3.2 Adding Participant: REFER to the Conference Focus*.

[5] BroadWorks also creates a bridge when a user invokes Shared Call Appearance (SCA) Bridging. For information about SCA Bridging, see the *BroadWorks Shared Call Appearance Interface Specification*.
[6] BroadWorks may also create an ad hoc conference when a user invokes the Directed Call Pickup with Barge-In (DPUBI) service.

©2019 CISCO SYSTEMS, INC.

RFC 4579 Section	Scenario Title	Cisco BroadWorks Support
5.5	REFER: Requesting a Focus to Add a New Resource to a Conference (Dial Out to a New Participant)	Cisco BroadWorks does not support this scenario.
5.6	REFER: Requesting a User to Dial in to a Conference Using a Conference URI	Cisco BroadWorks supports the concept behind this scenario for ad hoc conferences, provided that the participant who sends the REFER request is the conference creator. Specifically, the creator of an ad hoc conference can send a REFER request to bring another participant into the conference. Since Cisco BroadWorks is a B2BUA, the creator's user agent sends the REFER request to Cisco BroadWorks, which handles it on behalf of the remote user agent. Since Cisco BroadWorks permits only in-dialog REFER requests, Cisco BroadWorks does not support the "out of band" REFER request, as mentioned in *RFC 4579*. Cisco BroadWorks sends the *isfocus* parameter and the "conference" event package name in the *Allow-Events* header only to the conference creator. See the call flow in section *3.19.3.1 Adding a Participant: REFER Request to the User*.
5.7	REFER with REFER: Requesting a Focus to Refer a Participant to Dial in to the Conference	Cisco BroadWorks does not support this scenario.
5.8	Join Header Field: Dialing in to a Conference Using a (3rd Party) Dialog Identifier	Cisco BroadWorks does not support this scenario.
5.9	Replaces Header Field: Switching User Agents within a Conference	Cisco BroadWorks does not support this scenario.
5.10	Replaces Header Field: Transferring a Point-to-Point Session into a Conference	Cisco BroadWorks supports this scenario for ad hoc conferences, provided that the participant who sends the REFER request is the conference creator. Since Cisco BroadWorks is a B2BUA, it reuses the existing dialog with the remote user agent, sending a re-INVITE request to join the user agent to the conference mixer. Cisco BroadWorks sends the *isfocus* parameter and the *Allow-Events* header only to the conference creator. See the call flow in section *3.19.3.2 Adding Participant: REFER to the Conference Focus*.
5.11	REFER with BYE: Requesting That the Focus Remove a Participant from a Conference	Cisco BroadWorks supports this scenario for ad hoc conferences, provided that the participant who sends the REFER request is the conference creator.
5.12	Deleting a Conference	Cisco BroadWorks supports this scenario.
5.13	Discovery of URI Properties Using OPTIONS	Cisco BroadWorks does not support this scenario.

The Application Server can also perform the role of a Conference Notification Service, as explained in section *3.20 Session Initiation Protocol (SIP) Event Package for Conference State (RFC 4575)*.

©2019 CISCO SYSTEMS, INC.

3.19.2 Call Control for Ad Hoc Conferences

Because a Meet-Me conference is controlled by non-SIP means, the remainder of this section describes the SIP features Cisco BroadWorks supports for ad hoc conferences[7].

Any Cisco BroadWorks user can create an ad hoc conference, provided the user's service profile allows it. To create an ad hoc conference via SIP, the user agent sends a new INVITE request to the provisioned conference URI. When Cisco BroadWorks receives this INVITE request, it allocates a conference bridge and begins performing as a conference focus. Cisco BroadWorks then generates a conference ID, which it returns to the user agent as the URI in the *Contact* header. The conference ID has the same user part as the conference URI, but has a host and port as provisioned for the access-side interface. Cisco BroadWorks also adds an *isfocus* parameter to the *Contact* header and adds an *Allow-Events* header with the value "conference". For example call flows, see section *3.19.3.1 Adding a Participant: REFER Request to the User* and section *3.19.3.2 Adding Participant: REFER to the Conference Focus*.

As long as the ad hoc conference exists, the user agent that sent the initial INVITE remains the conference creator. The conference creator and Cisco BroadWorks can interact as conference-aware user agents. Other participants in the conference interact with Cisco BroadWorks as conference-unaware user agents.

After the conference creator creates the conference, it can add other user agents as participants to the conference. To add a participant using SIP, the creator sends a REFER request to Cisco BroadWorks. However, before it sends the REFER request, the creator must have an established call with the user it wants to add as a participant. The creator may send the REFER request to either the conference focus (see *3.19.3.2 Adding Participant: REFER to the Conference Focus*) or to the other user (see section *3.19.3.1 Adding a Participant: REFER Request to the User*). In either case, the creator must send the REFER request inside an existing dialog. Also note that in the latter case, the creator sends the REFER to Cisco BroadWorks inside the dialog associated with the call to the other user, and not directly to the other user's device. See the following subsection for details of these call scenarios.

The conference creator can also remove a participant using SIP. To remove a participant, the creator sends a REFER request with a method=BYE parameter in the *Refer-To* header. For an example call flow, see section *3.19.3.3 Removing a Participant: REFER Request with BYE*.

3.19.3 Conference Call Control Call Flows

This section contains example call flows that show how Cisco BroadWorks supports conference call control.

Note that in these example call flows, Cisco BroadWorks sends a NOTIFY request when it processes the REFER request from the access device. This behavior is configurable and is enabled by default. For more information on REFER request processing, see section *3.18 Session Initiation Protocol (SIP) Refer Method (RFC 3515)/SIP "Replaces" Header (RFC 3891)/SIP Referred-BY Mechanism (RFC 3892)*.

[7] BroadWorks also allows a user to control an ad hoc conference via client interfaces, such as the Xtended Services Interface (Xsi). Using these client interfaces, a user can create a conference, add a user to the conference, remove a user from the conference, and so on. The creator can then be a conference-unaware user agent, as BroadWorks requires only baseline SIP as specified in *RFC 3261*.

©2019 CISCO SYSTEMS, INC.

3.19.3.1 Adding a Participant: REFER Request to the User

The conference creator can add a participant to a conference by sending a REFER request to the user agent it wants to add. The creator sends the REFER request inside the dialog associated with the creator's call with the user. The *Refer-To* header must have the conference ID as its URI. This scenario is depicted in the following call flow.

The creator must use the conference ID as the *Refer-To* URI. Recall that Cisco BroadWorks returns the conference ID as the URI in the *Contact* header. If the creator uses the provisioned conference URI instead, Cisco BroadWorks rejects the REFER request.

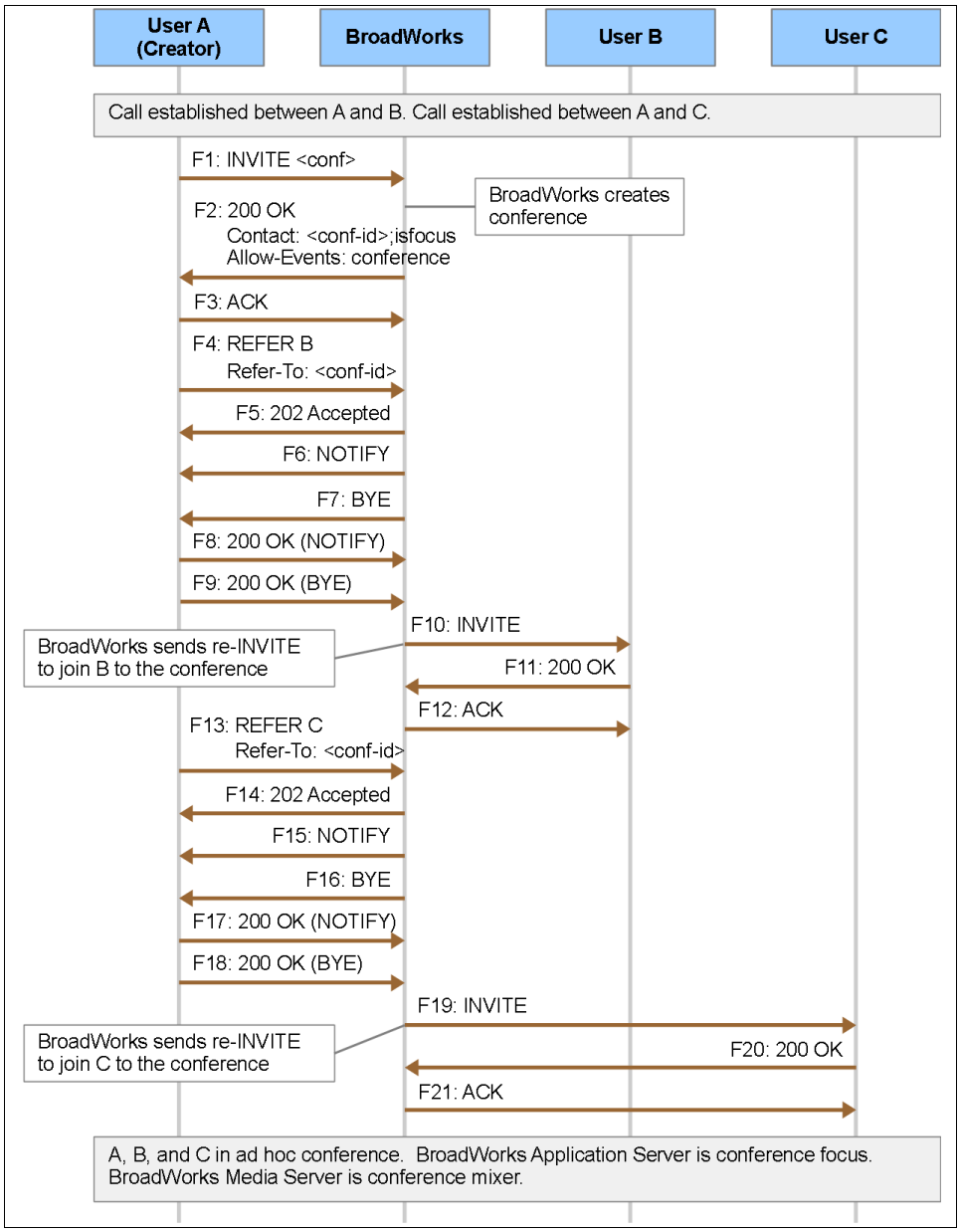

Figure 40 Call Flow Diagram for Adding a Participant to a Conference REFER Request to User

©2019 CISCO SYSTEMS, INC.

When the scenario begins, User A has an existing call with User B and User C. In this scenario, User A creates an ad hoc conference and adds User B and User C to the conference.

The SIP messaging begins with User A (the creator) sending an initial INVITE request (F1) to the conference URI.

F1: INVITE request from User A to Cisco BroadWorks

```
INVITE sip:focus@initech.test SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-6780966829
From: Bill Lumbergh <sip:101@initech.test>;tag=3020424457
To: <sip:focus@initech.test>
Call-ID: 1200434505
CSeq: 1 INVITE
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Content-Type: application/sdp
Content-Length: 153

(SDP omitted)
```

When Cisco BroadWorks receives the INVITE request, it creates the conference bridge and begins performing as the conference focus. Cisco BroadWorks sends a *200 OK* response (F2), which has the conference ID as the *Contact* URI, an *isfocus* parameter in the *Contact* header, and a "conference" event package name in the *Allow-Events* header.

F2: 200 response from Cisco BroadWorks to User A

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-6780966829
From:"Bill Lumbergh"<sip:101@initech.test>;tag=3020424457
To:<sip:focus@initech.test>;tag=1108304627-1380832219909
Call-ID:1200434505
CSeq:1 INVITE
Supported:
Contact:<sip:focus@10.16.145.3:5060>;isfocus
Call-Info:<sip:10.16.145.3>;appearance-index=3
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Accept:application/media_control+xml,application/sdp,multipart/mixed
Allow-Events:conference
Content-Type:application/sdp
Content-Length:139

(SDP omitted)
```

To add User B as a participant, User A sends a REFER request (F4) to Cisco BroadWorks inside the dialog established for User A's call with User B. The REFER request has the conference ID as the URI in the *Refer-To* header.

F4: REFER request from User A to Cisco BroadWorks

```
REFER sip:10.16.145.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-5460851929
From: <sip:101@initech.test>;tag=7395401105
To: <sip:102@initech.test>;tag=2014649204-1380832202245
Call-ID: 8841042227
CSeq: 3 REFER
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
```


©2019 CISCO SYSTEMS, INC.

```
Refer-To: <sip:focus@10.16.145.3:5060>
Content-Length: 0
```

Because Cisco BroadWorks is a B2BUA, Cisco BroadWorks does not forward this REFER request to User B, but handles the REFER itself. To connect User B to the conference mixer, Cisco BroadWorks sends a re-INVITE request (F10) to User B with the appropriate SDP. Cisco BroadWorks interacts with User B as a conference-unaware participant; therefore, the INVITE request does not have the *isfocus* parameter or the "conference" event package name in the *Allow-Events* header.

F10: INVITE request from Cisco BroadWorks to User B

```
INVITE sip:102@10.16.145.11:5060 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.11V5060-0-108272-162966048-1380832202203-
From:"Bill Lumbergh"<sip:101@initech.test;user=phone>;tag=162966048-
1380832202203-
To:"Dom Portwood"<sip:102@initech.test>;tag=a0cce773576dc85i0
Call-ID:BW153002203031013152508814@10.16.145.3
CSeq:108272 INVITE
Contact:<sip:10.16.145.3:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Supported:
Accept:application/media_control+xml,application/sdp,multipart/mixed
Max-Forwards:10
Content-Type:application/sdp
Content-Length:463

(SDP omitted)
```

To add User C as a participant, User A sends a REFER request (F13) to Cisco BroadWorks inside the dialog established for User A's call with User C. The SIP messages for adding User C are nearly identical to the SIP messages for adding User B and are omitted from this section.

©2019 CISCO SYSTEMS, INC.

3.19.3.2 Adding Participant: REFER to the Conference Focus

The conference creator can add a participant to a conference by sending a REFER request to the conference focus. The creator sends the REFER request inside the dialog associated with the creator's call to the conference focus. The REFER request has a *Refer-To* URI that identifies the user to add. The *Refer-To* URI also has an embedded *Replaces* header that identifies User A's dialog with the participant. This scenario is depicted in the following call flow.

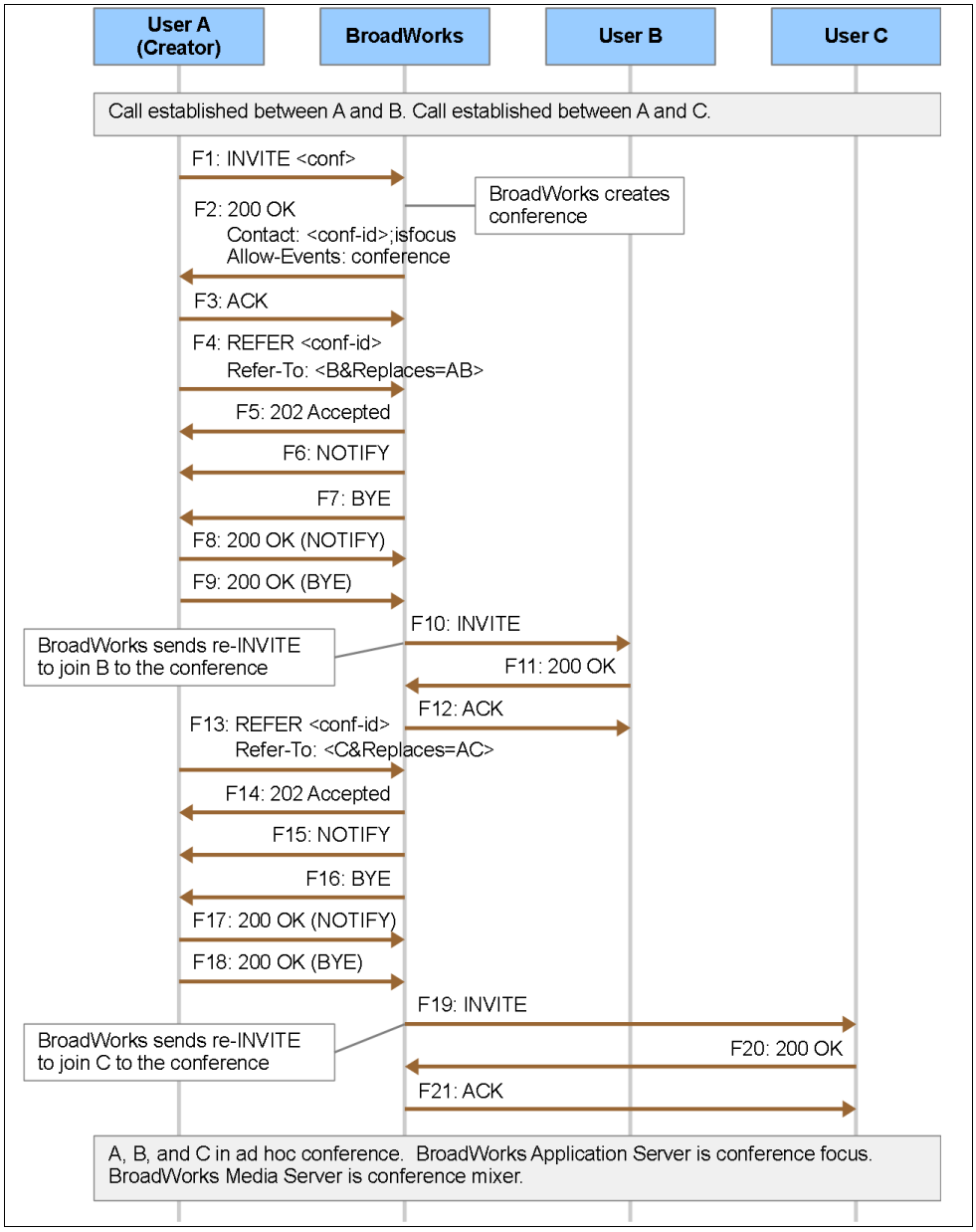

Figure 41 Call Flow Diagram for Adding a Participant to a Conference, REFER Request to Focus

When the scenario begins, User A has an existing call with User B and User C. In this scenario, User A creates an ad hoc conference and adds User B and User C to the conference.

©2019 CISCO SYSTEMS, INC.

The SIP messaging begins with User A (the creator) sending an initial INVITE request (F1) to the conference URI.

F1: INVITE request from User A to Cisco BroadWorks

```
INVITE sip:focus@initech.test SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-9271648669
From: Bill Lumbergh <sip:101@initech.test>;tag=6772925107
To: <sip:focus@initech.test>
Call-ID: 0037123978
CSeq: 1 INVITE
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Content-Type: application/sdp
Content-Length: 153

(SDP omitted)
```

When Cisco BroadWorks receives the INVITE request, it creates the conference bridge and begins performing as the conference focus. Cisco BroadWorks sends a *200 OK* response (F2), which has the conference ID as the *Contact* URI, an *isfocus* parameter in the *Contact* header, and a "conference" event package name in the *Allow-Events* header.

F2: 200 response from Cisco BroadWorks to User A

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-9271648669
From:"Bill Lumbergh"<sip:101@initech.test>;tag=6772925107
To:<sip:focus@initech.test>;tag=1730160321-1380833062850
Call-ID:0037123978
CSeq:1 INVITE
Supported:
Contact:<sip:focus@10.16.145.3:5060>;isfocus
Call-Info:<sip:10.16.145.3>;appearance-index=3
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Accept:application/media_control+xml,application/sdp,multipart/mixed
Allow-Events:conference
Content-Type:application/sdp
Content-Length:139

(SDP omitted)
```

To add User B as a participant, User A sends a REFER request (F4) to Cisco BroadWorks inside the dialog established for User A's call with the conference focus. The *Refer-To* URI identifies User B and has an embedded *Replaces* header that identifies the dialog between User A and User B's call session in Cisco BroadWorks.

F4: REFER request from User A to Cisco BroadWorks

```
REFER sip:focus@10.16.145.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-8452992605
From: <sip:101@initech.test>;tag=6772925107
To: <sip:focus@initech.test>;tag=1730160321-1380833062850
Call-ID: 0037123978
CSeq: 2 REFER
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Refer-To: <sip:102@initech.test?Replaces=4920859543%3Bfrom-
tag%3D5441092957%3Bto-tag%3D1484611295-1380833043509>
Referred-By: <sip:101@initech.test>
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

From the dialog identifier in the *Replaces* header, Cisco BroadWorks identifies User B's call session. To connect User B to the conference mixer, Cisco BroadWorks sends a re-INVITE request (F10) to User B with the appropriate SDP. Cisco BroadWorks interacts with User B as a conference-unaware participant; therefore, the INVITE request does not have the *isfocus* parameter or the "conference" event package name in the *Allow-Events* header.

F10: INVITE request from Cisco BroadWorks to User B

```
INVITE sip:102@10.16.145.11:5060 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.11V5060-0-528911-357809440-1380833043480-
From:"Bill Lumbergh"<sip:101@initech.test;user=phone>;tag=357809440-
1380833043480-
To:"Dom Portwood"<sip:102@initech.test>;tag=cc32e5e5ac83e986i0
Call-ID:BW154403480031013-1814963160@10.16.145.3
CSeq:528911 INVITE
Contact:<sip:10.16.145.3:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Supported:
Accept:application/media_control+xml,application/sdp,multipart/mixed
Max-Forwards:10
Content-Type:application/sdp
Content-Length:463

(SDP omitted)
```

To add User C as a participant, User A sends a REFER request (F13) to Cisco BroadWorks inside the dialog established for User A's call with the focus. The SIP messages for adding User C are nearly identical to the SIP messages for adding User B and are omitted from this section.

©2019 CISCO SYSTEMS, INC.

3.19.3.3 Removing a Participant: REFER Request with BYE

The conference creator can remove a participant from the conference by sending a REFER request to the conference focus. The creator sends the REFER request inside the dialog associated with the creator's call to the conference focus. The REFER request has a *Refer-To* URI that identifies the user to remove and a *method* parameter that contains the value "BYE". This scenario is depicted in the following call flow.

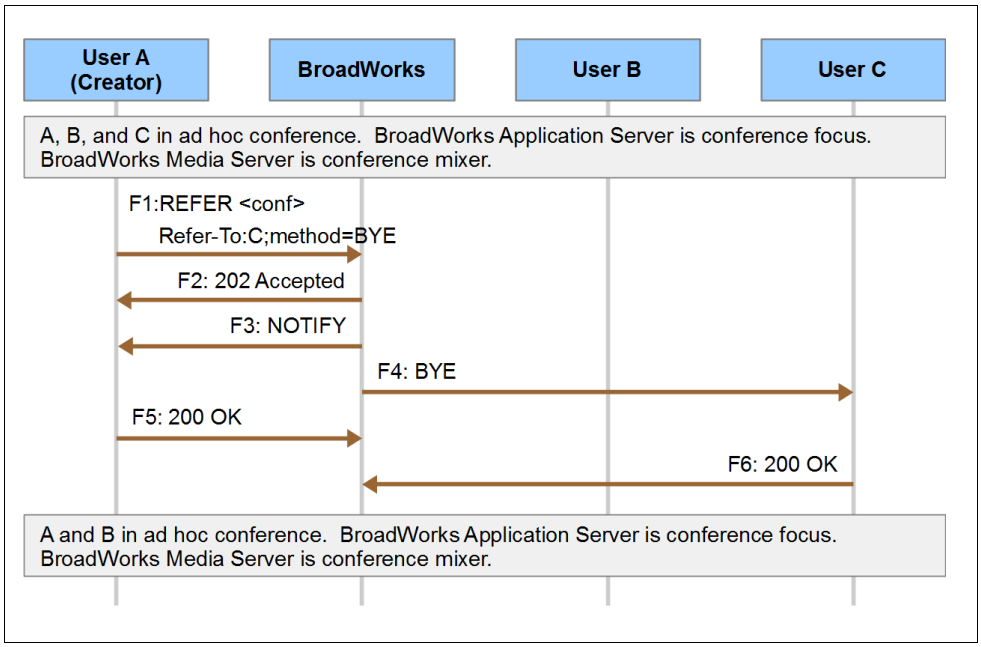

Figure 42 Call Flow Diagram for Removing a Participant from a Conference

When the scenario begins, User A has an existing conference with User C as a participant.

To remove User C from the conference, User A sends a REFER request (F1) to Cisco BroadWorks inside the dialog established for User A's call with the conference focus. The *Refer-To* URI identifies User C and has a *method* with the value BYE. To identify User C, the *Refer-To* URI has User C's directory number (used in this example) or user ID, which User A obtained either from the conference events notification or from non-SIP means.

F1: REFER request from User A to Cisco BroadWorks

```
REFER sip:focus@10.16.145.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-8120492382
From: <sip:101@initech.test>;tag=5212969979
To: <sip:focus@initech.test>;tag=1848268181-1380833191520
Call-ID: 4291450576
CSeq: 2 REFER
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Refer-To: <sip:5125550103@initech.test;method=BYE>
Content-Length: 0
```

When Cisco BroadWorks receives the REFER request, it identifies User C from the *Refer-To* URI, then sends a BYE request (F4) to User C and releases the call.

©2019 CISCO SYSTEMS, INC.

F4: BYE request from Cisco BroadWorks to User C

```
BYE sip:103@10.16.145.11:5060 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.11V5060-0-598182-332442945-1380833182020-
From:"Bill Lumbergh"<sip:101@initech.test;user=phone>;tag=332442945-
1380833182020-
To:"Peter Gibbons"<sip:103@initech.test>;tag=32c1bf4e3c7c6d57i1
Call-ID:BW154622020031013204015640 6@10.16.145.3
CSeq:598182 BYE
Max-Forwards:10
Content-Length:0
```


©2019 CISCO SYSTEMS, INC.

3.20 Session Initiation Protocol (SIP) Event Package for Conference State (RFC 4575)

3.20.1 Procedures

Cisco BroadWorks supports the conference state event package, which permits participants in a conference to receive information about the conference itself as well as information about the other participants in the conference.

Cisco BroadWorks supports two types of conferences, and the type of conference affects Cisco BroadWorks behavior with regard to the conference event package. One type is an ad hoc conference, which uses a transient conference bridge. A Cisco BroadWorks user may create an ad hoc conference by an invocation of one of these services: Three-Way Call, N-Way Call, and Directed Call Pickup Barge-in (DPUBI). The other type of conference is a Meet-Me conference, which uses a provisioned conference bridge.

To subscribe to the conference events, an access device must send a SUBSCRIBE request to the Application Server. The request must have the conference subscription URI as the *Request-URI* and an *Event* header with the value "conference". If the SUBSCRIBE request has an *Accept* header, it must contain the media type application/conference-info+xml. If the request does not have an *Accept* header, then Cisco BroadWorks assumes application/conference-info+xml as the default.

Cisco BroadWorks does not provide a single conference subscription URI for a conference. Instead, it creates a unique conference subscription URI for each participant who is allowed to subscribe to the conference events. Moreover, the participant's conference subscription URI is valid only as long as the participant is in the conference. A participant may create multiple subscriptions, but Cisco BroadWorks allows only one subscription per device endpoint. Therefore, to create multiple subscriptions, a participant must have multiple device endpoints configured via Shared Call Appearance (SCA).

RFC 4579 describes how the conference focus should send the conference subscription URI as the *Contact* header URI with an *isfocus* parameter. Additionally, the focus should send the *Allow-Events* header with the value "conference". Cisco BroadWorks, acting as the conference focus, follows these requirements when interacting with the conference bridge creator for an ad hoc conference[8]. When a Cisco BroadWorks user sends an INVITE request to the conference URI, Cisco BroadWorks sends a 200 response with *Contact* header and *Allow-Events* header as required by *RFC 4579* (provided no other condition occurs to force a different response). The *Contact* header contains the conference subscription URI and an *isfocus* parameter.

[8] In the case of DPUBI, the conference creator dials a feature access code, and not the conference focus URI. This is different from a Three-Way Call or an N-Way Call, in which the conference creator directly calls the conference URI. Consequently, BroadWorks does not follow *RFC 4579* for DPUBI, that is, it does not send the *isfocus* parameter or the *Allow-Events* header.

©2019 CISCO SYSTEMS, INC.

For all other participants in a conference who are allowed to subscribe to the conference events, Cisco BroadWorks does not follow *RFC 4579* for advertising the conference subscription URI. Instead, Cisco BroadWorks provides the conference subscription URI in a NOTIFY request for the Call-Info event package. For a description of the Call-Info event package, see the *BroadWorks SIP Access Side Extensions Interface Specification* [43]. The NOTIFY request contains a *Call-Info* header, which contains a *conference-subscription-uri* parameter. The value of this parameter is the conference subscription URI defined uniquely for that participant. To receive the NOTIFY request, the device endpoint must subscribe to the Call-Info event package. (Sending the conference subscription URI in the Call-Info NOTIFY is useful, for example, if a user has a desk phone, with a basic display, as well as a soft client phone with an extended display. The user may join the conference with his desk phone, while still seeing all the conference participants in the soft client's display. The soft client requires the NOTIFY to receive the conference subscription URI.)

NOTE: The creator of the conference bridge for an ad hoc conference receives the conference subscription URI in the *200* response, as previously described. The creator may also have device endpoints that subscribe to the Call-Info event package. If the creator creates the conference via an INVITE request from the device endpoint (say, according to *RFC 4579* procedures), then Cisco BroadWorks does not send the conference subscription URI to these device endpoints via the Call-Info NOTIFY request. However, if the creator creates the conference through non-SIP means, such as via a client interface, then Cisco BroadWorks sends the conference subscription URI via the Call-Info NOTIFY to the creator's device endpoints.

Cisco BroadWorks only supports conference events with access devices and only with Cisco BroadWorks users who are in the same group or enterprise as the "owner" of the conference. For an ad hoc conference, the owner is the Cisco BroadWorks user who created the transient conference bridge. For a Meet-Me conference, the owner is the virtual subscriber for the service.

The conference events subscription exists until it terminates. Cisco BroadWorks supports the usual termination conditions:

- Cisco BroadWorks terminates the subscription when it expires. The subscriber must send a SUBSCRIBE request to refresh the subscription, as required by *RFC 6665*. If the subscription expires, Cisco BroadWorks silently terminates the subscription and does not send a NOTIFY request.
- Cisco BroadWorks terminates the subscription if the subscriber "unsubscribes" by sending a SUBSCRIBE request with the *Expires* header set to "0". In this case, Cisco BroadWorks sends a NOTIFY request with *Subscription-State* "terminated" and the *reason* parameter omitted.
- Cisco BroadWorks terminates the subscription when the subscriber leaves the conference. In this case, BroadWorks sends a NOTIFY request with *Subscription-State* "terminated" and *reason* set to "noresource".
- Cisco BroadWorks terminates an existing subscription if it receives a SUBSCRIBE for a new subscription for the same device endpoint. In this case, Cisco BroadWorks silently terminates the previously existing subscription and does not send a NOTIFY request.

©2019 CISCO SYSTEMS, INC.

3.20.2 Conference Event Information

RFC 4575 describes the set of conference information that may be provided in the body of the SUBSCRIBE request. Most of this information is optional. Cisco BroadWorks provides the following information:

- Conference description
- Conference state
- Conference participants

The following diagram presents the overall hierarchy of the conference-info XML schema supported by this feature.

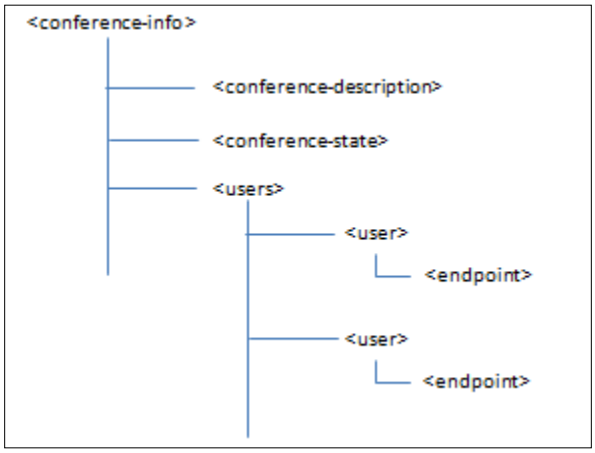

Figure 43 Conference-info XML Schema Hierarchy

The following tables describe in more detail the conference information Cisco BroadWorks provides.

<conference-info> attributes		
Name	Description	Example
entity	This attribute contains the conference URI that identifies the conference being described in the XML document. This is the SIP URI that an interested entity needs to SUBSCRIBE to get the conference package information. This is the conference subscription URI.	sip:confsub@example.com
state	This is the state of the XML document. It has three values: "full", "partial", and "deleted". Cisco BroadWorks sets the state to "full" for initial notification in response to the subscription. It sets the state to "partial" for subsequent notification as long as there is at least one participant in a Meet-Me conference or at least two participants in an ad hoc conference. When the conference ceases to exist, the state is set to "deleted" and no child elements are included in the body.	full partial deleted
version	This is a 32-bit integer. It is incremented by one for each subsequent notification sent to the subscriber.	10

<conference-description> child elements

©2019 CISCO SYSTEMS, INC.

<conference-description> child elements		
Element	**Description**	**Example**
subject	Contains the subject of the conference. For a Meet-Me conference, this field contains the provisioned Meet-Me conference title. For a Three-way or N-way conference, this field is set to “Ad hoc”.	“Feature discussion”
maximum-user-count	For a Three-way/N-way conference, this is the total number of ports allocated for the conference. When the Three-way/N-way conferences are daisy-chained, this number does not represent the total number of ports allocated for each conference. For a Meet-Me conference, this is the “Estimated Participants” value if it is configured. If it is not configured, this value is set to “147”.	5

<conference-state> child elements		
Element	**Description**	**Example**
user-count	This value represents the overall number of users who joined the conference. This number does not necessarily match and may exceed the number of entries in the <users> container.	33
active	This is a Boolean element. When this element is present, the conference focus exists and may be able to join more participants to the conference as long as the conference still has available ports. When this element is not present, it means the conference focus is terminated.	active
locked	This is a Boolean element. This element is applicable for a Meet-Me conference. A moderator can lock or unlock the conference. When the Meet-Me conference is locked, a new user cannot join the conference, but the moderator can join a new user to the conference. State change from active to locked is notified to moderator(s) who is/are in the conference and not to the participants. When the moderator unlocks the conference, notification is sent to all moderators with the “active” element present in the <conference-state> element.	locked

<Users> container element and <user> sub-elements		
Attribute / Element	**Description**	**Example**

©2019 CISCO SYSTEMS, INC.

<Users> container element and <user> sub-elements		
entity attribute	This attribute contains the participant's identity. The 'entity' value is unique among all participants in the conference. If the participant is a user in the same group or enterprise, then the value is the participant's user ID (see example 1). Otherwise, the value is the participant's CLID, unless the participant has an anonymous identity (see example 2). If the participant is anonymous, then the value is "Anonymous_X" sip:anonymous_X@anonymous.invalid, where X is a unique identifier for the anonymous participant (see example 3). As long the anonymous participant is still a member of the conference, the participant is identified by this unique anonymous identifier. The unique anonymous identifier is released when the participant leaves the conference.	Example 1: joe@example.com Example 2: tel:+19726994601 Example 3: "Anonymous_3_1" sip:anonymous_3_1@anonymous.invalid
state attribute	This attribute is set to "full" when the state of the <conference-info> is "full" (initial notification). The <user> entry with "full" state means that the user is a member of this conference. A "deleted" value indicates that the user is no longer a participant of the conference. The "partial" value is not used by Cisco BroadWorks.	full deleted
display-text element	This attribute contains the user's first name and last name.	"Joe Smith"

©2019 CISCO SYSTEMS, INC.

<Users> container element and <user> sub-elements		
endpoint element	This is the device endpoint that the user uses to join the conference, that is, has a media session established with the conference focus. The <endpoint> element has its own attributes and elements. Cisco BroadWorks only includes the <status> element. Cisco BroadWorks supports the following status values: ▪ "Connected": The endpoint is a participant of the conference and has bi-directional voice path with the conference. Cisco BroadWorks reports this status to all members of a Three-Way, N-Way, or Meet-Me conference. ▪ "Disconnected": The endpoint is no longer a participant in the conference and the conference port that it used is returned to the conference focus. The "state" attribute of the <user> element is set to "deleted". Cisco BroadWorks reports this status to all members of a Three-Way, N-Way, or Meet-Me conference. ▪ "On-hold": The endpoint does not send/ receive media to/from the conference bridge. Cisco BroadWorks reports this status to the controller and not to the other participants of a Three-Way or N-Way conference. Cisco BroadWorks does not report this status to the participants of a Meet-Me conference. ▪ "Deaf": Active signaling dialog exists between the endpoint and the focus. The endpoint cannot listen to the conference but the endpoint media is mixed into the conference. Cisco BroadWorks reports this status to the controller but not to the other participants of a Three-Way or N-Way conference. Cisco BroadWorks reports this status to all participants of a Meet-Me conference. NOTE: This status is Cisco BroadWorks proprietary and is mapped to the 'connected' status. ▪ "Muted-via-focus": Active signaling dialog exists between the endpoint and the focus. The endpoint can listen to the conference but the endpoint's media is not mixed into the conference. Cisco BroadWorks reports this status to the controller but not to the other participants of a Three-Way or N-Way conference. Cisco BroadWorks reports this status to all members of a Meet-Me conference.	Connected Disconnected On-Hold Muted-via-focus

©2019 CISCO SYSTEMS, INC.

3.20.3 Conference Subscription Call Flows

3.20.3.1 Subscribing to Conference Events

In the following scenario, a Cisco BroadWorks user has two device endpoints. Device A-1 is the user's primary device endpoint, which can be a desk phone. Device A-2 is a Shared Call Appearance device endpoint, which can be, for example, a soft phone client device running on a personal computer. Device A-2 has a more advanced display, and therefore it subscribes to the conference events and displays the conference information it receives. Since Device A-2 is not a participant in the conference, it learns the conference subscription URI via the call-info events, as explained in the *BroadWorks SIP Access Side Extensions Interface Specification* [43].

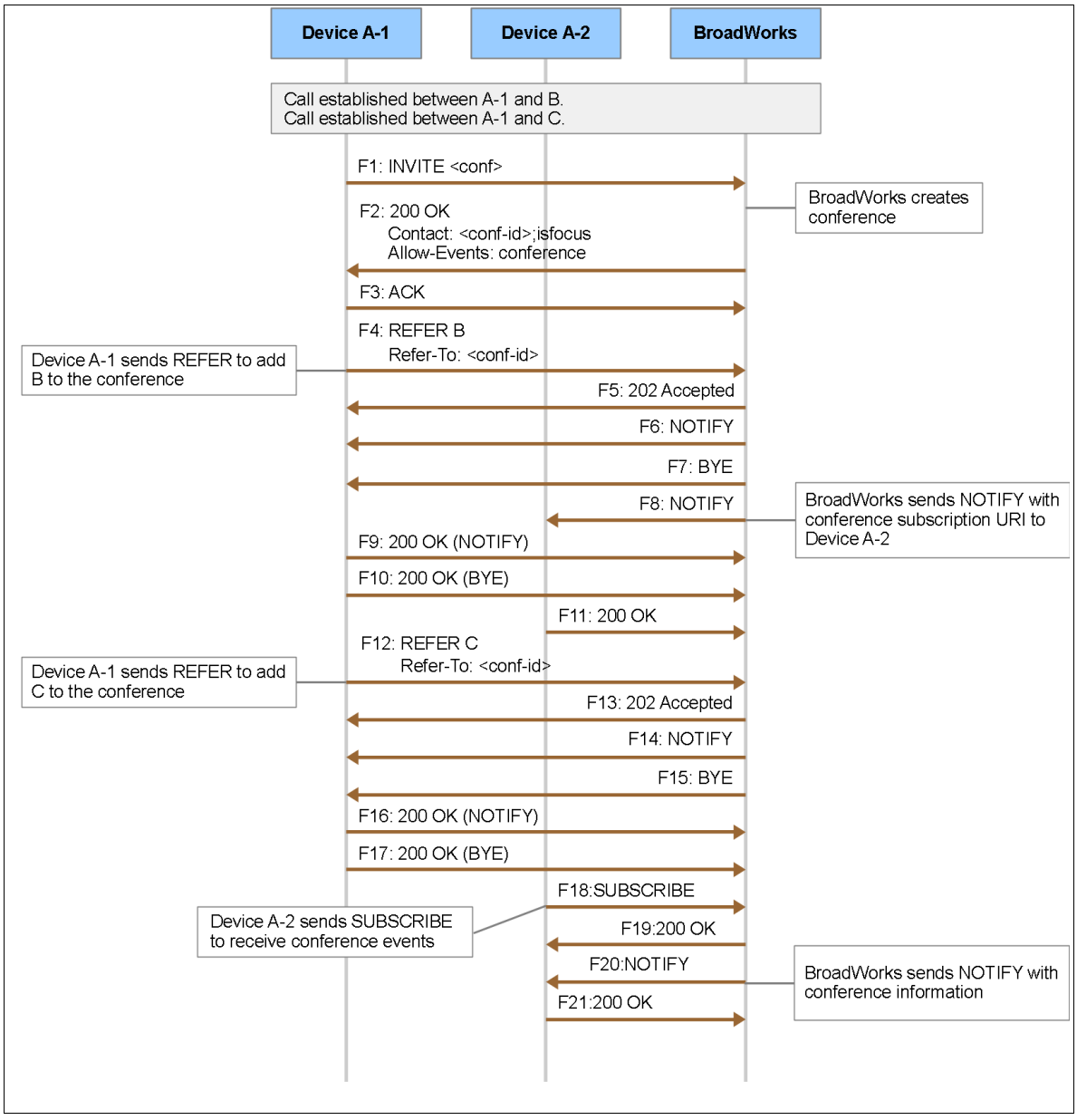

Figure 44 Call Flow Diagram for Subscribing to Conference Events

©2019 CISCO SYSTEMS, INC.

When this scenario begins, Device A-1 already has established calls with User A and User B.

The call flow begins with Device A-1 sending an initial INVITE request (F1) to the conference URI to create the ad hoc conference bridge.

F1: INVITE request from Device A-1 to Cisco BroadWorks

```
INVITE sip:focus@initech.test SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-8967270347
From: Bill Lumbergh <sip:101@initech.test>;tag=5271426369
To: <sip:focus@initech.test>
Call-ID: 3856325981
CSeq: 1 INVITE
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Content-Type: application/sdp
Content-Length: 152

(SDP omitted)
```

BroadWorks sends a *200* response (F2), which has the conference ID as the URI in the *Contact* header. The contact header also has the *isfocus* feature parameter. Device A-1 (and only Device A-1) can subscribe to this URI to receive conference events.

F2: 200 response from Cisco BroadWorks to Device A-1

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-8967270347
From:"Bill Lumbergh"<sip:101@initech.test>;tag=5271426369
To:<sip:focus@initech.test>;tag=88942611-1380926017826
Call-ID:3856325981
CSeq:1 INVITE
Supported:
Contact:<sip:focus@10.16.145.3:5060>;isfocus
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Accept:application/media_control+xml,application/sdp,multipart/mixed
Allow-Events:conference
Content-Type:application/sdp
Content-Length:139

(SDP omitted)
```

After Device A-1 creates the conference bridge, it sends a REFER request (F4) to Cisco BroadWorks to add User B to the conference. Device A-1 sends this REFER in the dialog that is associated with its established call with User B. (Note that the SIP messaging between Cisco BroadWorks and User B is not shown.)

F4: REFER request from Device A-1 to Cisco BroadWorks

```
REFER sip:10.16.145.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-8909276848
From: <sip:101@initech.test>;tag=2957079164
To: <sip:102@initech.test>;tag=230410464-1380925996578
Call-ID: 8947181181
CSeq: 3 REFER
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Refer-To: <sip:focus@10.16.145.3:5060>
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

Cisco BroadWorks sends a NOTIFY request (F8) for the call-info package to Device A-2. The *conference-subscription-uri* parameter in the *Call-Info* header provides the conference subscription URI to Device A-2. Device A-2 may use this URI to subscribe to the conference events.

F8: NOTIFY request from Cisco BroadWorks to Device A-2

```
NOTIFY sip:102a@10.16.145.5:5090 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.5V5090-0-94032173-1233342373-1380925917172
From:<sip:102a@initech.test>;tag=1233342373-1380925917172
To:"Dom Portwood"<sip:102a@initech.test>;tag=6322771492
Call-ID:3337398434
CSeq:94032173 NOTIFY
Contact:<sip:10.16.145.3:5060>
Call-Info:<sip:10.16.145.3>;appearance-state=active;appearance-
uri="\"Bill Lumbergh\"<sip:101@initech.test;user=phone>";appearance-
index=1;conference-subscription-uri="<sip:etid=381b42ca-8acb-490c-ba4b-
310f1b1aecca@10.16.145.3>",<sip:10.16.145.3>;appearance-
state=idle;appearance-index=*
Event:call-info
Subscription-State:active;expires=499
Max-Forwards:10
Content-Length:0
```

Device A-1 sends a REFER request (F12) to Cisco BroadWorks to add User C to the conference. Device A-1 sends this REFER in the dialog that is associated with its established call with User C. (Note that the SIP messaging between Cisco BroadWorks and User C is not shown.)

F12: REFER request from Device A-1 to Cisco BroadWorks

```
REFER sip:10.16.145.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-2765440954
From: <sip:101@initech.test>;tag=2653803870
To: <sip:103@initech.test>;tag=467944300-1380926005441
Call-ID: 3549690172
CSeq: 3 REFER
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Refer-To: <sip:focus@10.16.145.3:5060>
Content-Length: 0
```

Device A-2 sends a SUBSCRIBE request (F18) to Cisco BroadWorks to subscribe to the conference events. For the *Request-URI*, Device A-2 uses the conference subscription URI that it received in the NOTIFY request (F8).

F18: SUBSCRIBE request from Device A-2 to Cisco BroadWorks

```
SUBSCRIBE sip:etid=381b42ca-8acb-490c-ba4b-310f1b1aecca@10.16.145.3
SIP/2.0
Via: SIP/2.0/UDP 10.16.145.5:5090;branch=z9hG4bK-2218546989
From: Dom Portwood <sip:102a@initech.test>;tag=3165944943
To: <sip:etid=381b42ca-8acb-490c-ba4b-310f1b1aecca@10.16.145.3>
Call-ID: 5059863090
CSeq: 1 SUBSCRIBE
Contact: <sip:102a@10.16.145.5:5090>
Max-Forwards: 70
Event: conference
```


©2019 CISCO SYSTEMS, INC.

```
Expires: 600
Content-Length: 0
```

Cisco BroadWorks responds immediately to the SUBSCRIBE request, sending a NOTIFY request (F20) with the conference information.

F20: NOTIFY request from Cisco BroadWorks to Device A-2

```
NOTIFY sip:102a@10.16.145.5:5090 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.5V5090-0-94035189-1205110106-1380926020849
From:<sip:etid=381b42ca-8acb-490c-ba4b-
310f1b1aecca@10.16.145.3>;tag=1205110106-1380926020849
To:"Dom Portwood"<sip:102a@initech.test>;tag=3165944943
Call-ID:5059863090
CSeq:94035189 NOTIFY
Contact:<sip:10.16.145.3:5060>
Event:conference
Subscription-State:active;expires=599
Max-Forwards:10
Content-Type:application/conference-info+xml
Content-Length:803

<?xml version="1.0" encoding="UTF-8"?>
<conference-info xmlns="urn:ietf:params:xml:ns:conference-info"
entity="sip:etid=381b42ca-8acb-490c-ba4b-310f1b1aecca@10.16.145.3"
state="full" version="1">
<conference-description>
<subject>Ad-hoc</subject>
<maximum-user-count>6</maximum-user-count>
</conference-description>
<conference-state>
<active/>
<user-count>3</user-count>
</conference-state>
<users>
<user entity="dportwood@initech.test" state="full">
<endpoint state="full">
<status>connected</status>
</endpoint>
</user>
<user entity="blumbergh@initech.test" state="full">
<endpoint state="full">
<status>connected</status>
</endpoint>
</user>
<user entity="pgibbons@initech.test" state="full">
<endpoint state="full">
<status>connected</status>
</endpoint>
</user>
</users>
</conference-info>
```


©2019 CISCO SYSTEMS, INC.

3.20.3.2 Receiving Conference Events

This scenario continues from the preceding scenario. User A, who has device endpoints Device A-1 and Device A-2, is in an ad hoc conference with User B and User C. Device A-1 is the controlling participant in the conference. Device A-2 is subscribed to the conference events. This scenario shows Cisco BroadWorks sending NOTIFY requests to provided updated conference information to the subscriber.

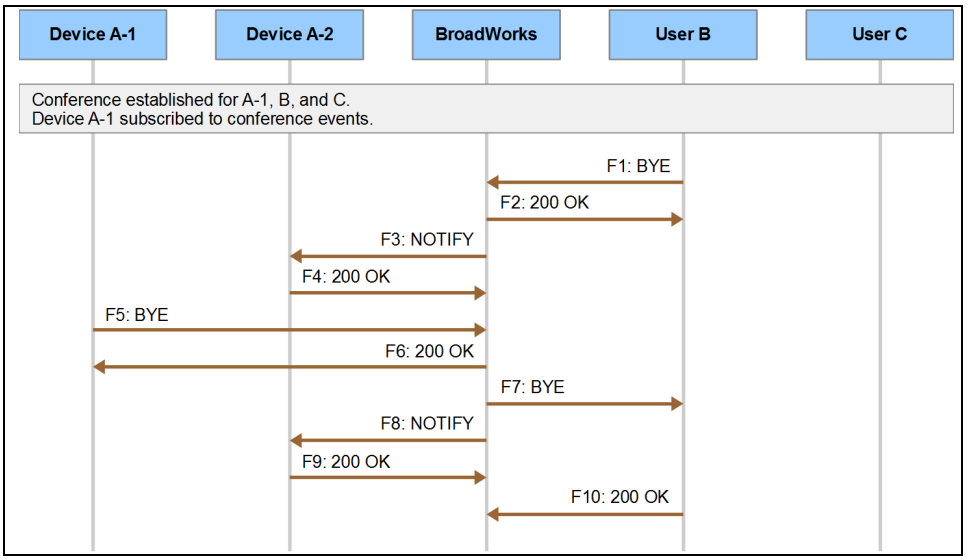

Figure 45 Call Flow Diagram for Receiving Conference Events

The scenario starts with User B sending a BYE request (F1) to leave the conference.

F1: BYE request from User B to Cisco BroadWorks

```
BYE sip:10.16.145.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.16.145.11:5060;branch=z9hG4bK-11b1469b
From: "Peter Gibbons" <sip:103@initech.test>;tag=864b20c4c203ce6i1
To: "Bill Lumbergh" <sip:101@initech.test;user=phone>;tag=644165093-
1380926005400-
Call-ID: BW173325400041013-10967960 4@10.16.145.3
CSeq: 101 BYE
Max-Forwards: 70
Content-Length: 0
```

Cisco BroadWorks sends a NOTIFY request (F3) to Device A-2 with updated conference information.

F3: NOTIFY request from Cisco BroadWorks to Device A-2

```
NOTIFY sip:102a@10.16.145.5:5090 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.5V5090-0-94039022-1205110106-1380926020849
From:<sip:etid=381b42ca-8acb-490c-ba4b-
310f1b1aecca@10.16.145.3>;tag=1205110106-1380926020849
To:"Dom Portwood"<sip:102a@initech.test>;tag=3165944943
Call-ID:5059863090
CSeq:94039022 NOTIFY
Contact:<sip:10.16.145.3:5060>
Event:conference
Subscription-State:active;expires=596
Max-Forwards:10
```


©2019 CISCO SYSTEMS, INC.

```
Content-Type:application/conference-info+xml
Content-Length:367

<?xml version="1.0" encoding="UTF-8"?>
<conference-info xmlns="urn:ietf:params:xml:ns:conference-info"
entity="sip:etid=381b42ca-8acb-490c-ba4b-310f1b1aecca@10.16.145.3"
state="partial" version="2">
<users>
<user entity="sip:pgibbons@initech.test" state="full">
<endpoint state="full">
<status>disconnected</status>
</endpoint>
</user>
</users>
</conference-info>
```

User A leaves the conference, causing Device A-1 to send a BYE request (F5) to Cisco BroadWorks.

F5: BYE request from Device A-1 to Cisco BroadWorks

```
BYE sip:focus@10.16.145.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.16.145.6:5060;branch=z9hG4bK-5347427049
From: <sip:101@initech.test>;tag=5271426369
To: <sip:focus@initech.test>;tag=88942611-1380926017826
Call-ID: 3856325981
CSeq: 2 BYE
Contact: <sip:101@10.16.145.6:5060>
Max-Forwards: 70
Allow: ACK,BYE,CANCEL,INVITE,OPTIONS,PRACK,UPDATE
Content-Length: 0
```

Cisco BroadWorks shuts down the conference. When it does so, it sends a BYE request (F7) to User C.

F7: BYE request from Cisco BroadWorks to User C

```
BYE sip:102@10.16.145.11:5060 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.11V5060-0-47005441-1193872240-1380925996538-
From:"Bill Lumbergh"<sip:101@initech.test;user=phone>;tag=1193872240-
1380925996538-
To:"Dom Portwood"<sip:102@initech.test>;tag=14c0637612ab8665i0
Call-ID:BW173316538041013160864355 5@10.16.145.3
CSeq:47005441 BYE
Max-Forwards:10
Content-Length:0
```

Cisco BroadWorks sends a final NOTIFY request (F8) to Device A-2. Cisco BroadWorks adds a *Conference-State* header with the value “terminated” and a *reason* parameter set to “noresource”. The XML document in the request body indicates a “deleted” state for the conference.

F8: NOTIFY request from Cisco BroadWorks to Device A-2

```
NOTIFY sip:102a@10.16.145.5:5090 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.5V5090-0-94063212-1205110106-1380926020849
From:<sip:etid=381b42ca-8acb-490c-ba4b-
310f1b1aecca@10.16.145.3>;tag=1205110106-1380926020849
```


©2019 CISCO SYSTEMS, INC.

```
To:"Dom Portwood"<sip:102a@initech.test>;tag=3165944943
Call-ID:5059863090
CSeq:94063212 NOTIFY
Contact:<sip:10.16.145.3:5060>
Event:conference
Subscription-State:terminated;reason=noresource
Max-Forwards:10
Content-Type:application/conference-info+xml
Content-Length:219

<?xml version="1.0" encoding="UTF-8"?>
<conference-info xmlns="urn:ietf:params:xml:ns:conference-info"
entity="sip:etid=381b42ca-8acb-490c-ba4b-310f1b1aecca@10.16.145.3"
state="deleted" version="3">
</conference-info>
```


©2019 CISCO SYSTEMS, INC.

3.21 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 access interface:

- Call-info (Cisco BroadWorks proprietary event package for support of enhanced shared call appearances for key system emulation and enhanced business applications. For more information, see the *BroadWorks SIP Access Side Extensions Interface* [43].
- Line-Seize (Cisco BroadWorks proprietary event package for support of enhanced shared call appearances for key system emulation and enhanced business applications). For more information, see the *BroadWorks SIP Access Side Extensions Interface* [43].
- Message-summary (message-summary). For more information, see the *A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)* [17].

Support for additional event packages will be added to Cisco BroadWorks in the future, enabling a wide variety of service solutions.

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.

©2019 CISCO SYSTEMS, INC.

3.22 Message Summary and Message Waiting Indication Event Package for Session Initiation Protocol (SIP) (RFC 3842)

Cisco BroadWorks supports this functionality:

Cisco BroadWorks can operate in two modes depending on device settings.

- In the first mode, Cisco BroadWorks requires that the device subscribes to the event package to receive NOTIFY requests.
- In the second mode, the subscription by the device is optional.

When it receives a SUBSCRIBE request for the message summary event package, Cisco BroadWorks executes the following logic:

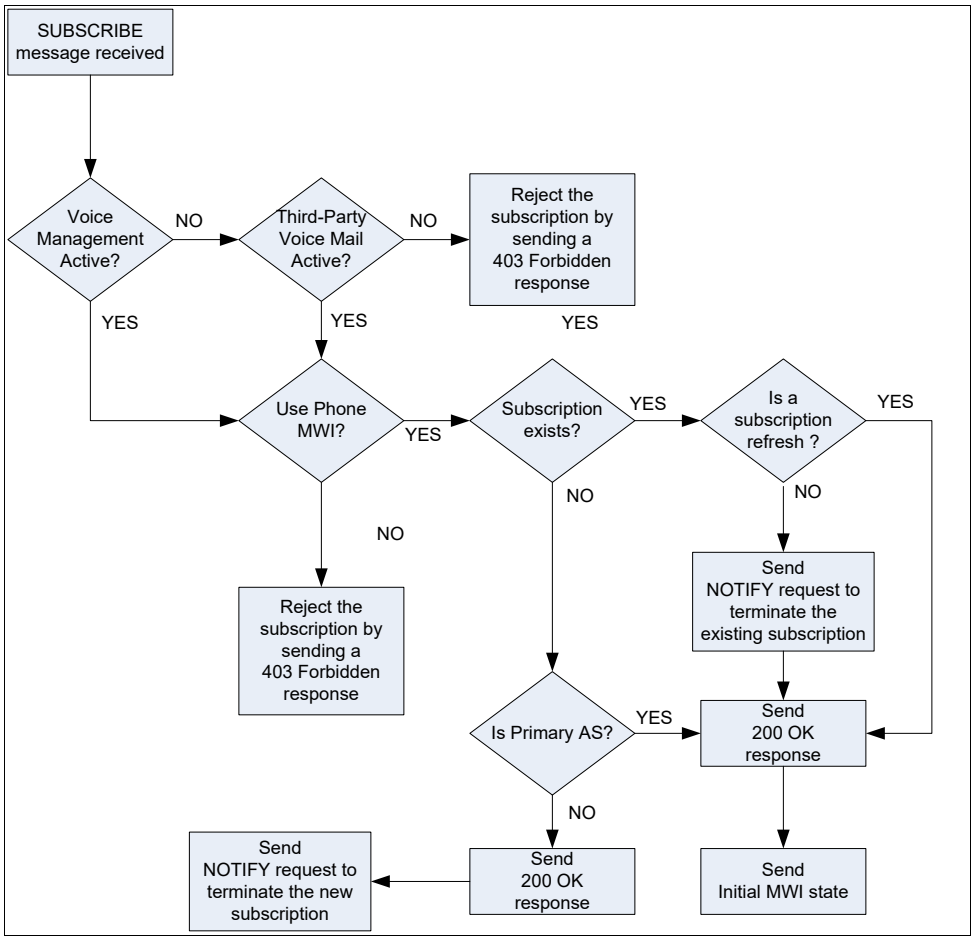

Figure 46 SUBSCRIBE Request Processing Flowchart

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.

©2019 CISCO SYSTEMS, INC.

Figure 47 shows the signaling flow for an accepted subscription.

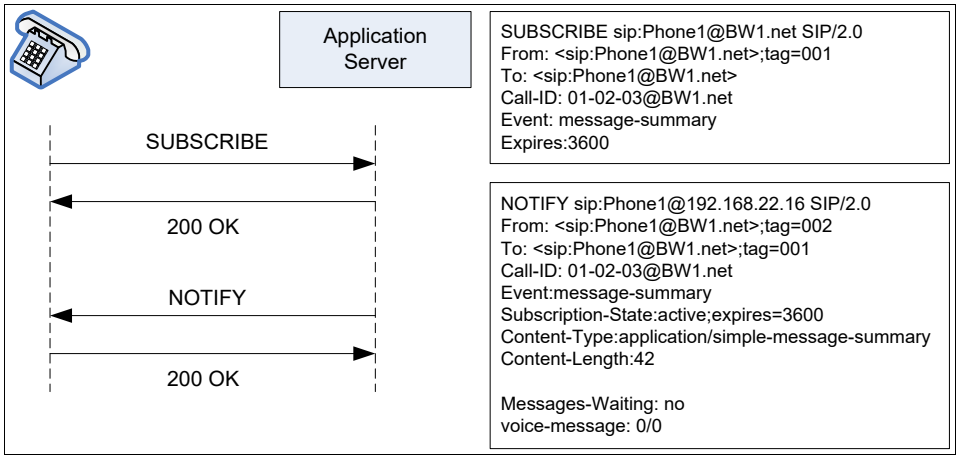

Figure 47 Subscription to Message-summary

If the access device does not subscribe to the message-summary event package, Cisco BroadWorks can send a NOTIFY request with the state terminated dependent on the device configuration. This is shown in the *Figure 48*.

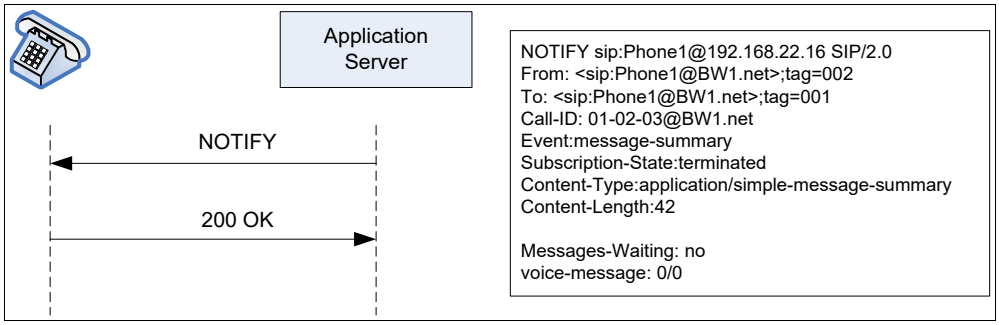

Figure 48 Unsubscribed Message-summary Notification

After receiving a NOTIFY method with the message-summary *Event* header and application/simple-message-summary body, an access device should provide Stutter Dial Tone instead of regular dial tone to a user who goes off hook. The access device should also provide a visual message waiting indicator to the user upon receipt of the NOTIFY method.

For devices that do not SUBSCRIBE to the message-summary event package, Cisco BroadWorks optionally sends a NOTIFY request when a device registers to refresh its Voice Message Waiting Indication. This is not required for subscribed devices because the subscription acts as a refresh request. The refresh on register is configured from the CLI.

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

To prevent excessive NOTIFY requests, multiple registrations in a short period of time generate a single NOTIFY request. This time interval is configured from the CLI.

```
AS_CLI/Service/VoiceMessageSummaryUpdate> set
minTimeBetweenMWIOnRegisterInSeconds 900
```


©2019 CISCO SYSTEMS, INC.

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

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


©2019 CISCO SYSTEMS, INC.

3.23 Session Initiation Protocol (SIP) Extension for Instant Messaging (RFC 3428)

Cisco BroadWorks supports Short Message Service handling using MESSAGE according to *RFC 3428*. This service allows text messages to be sent from:

- 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 from and to other protocols, when needed.

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

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

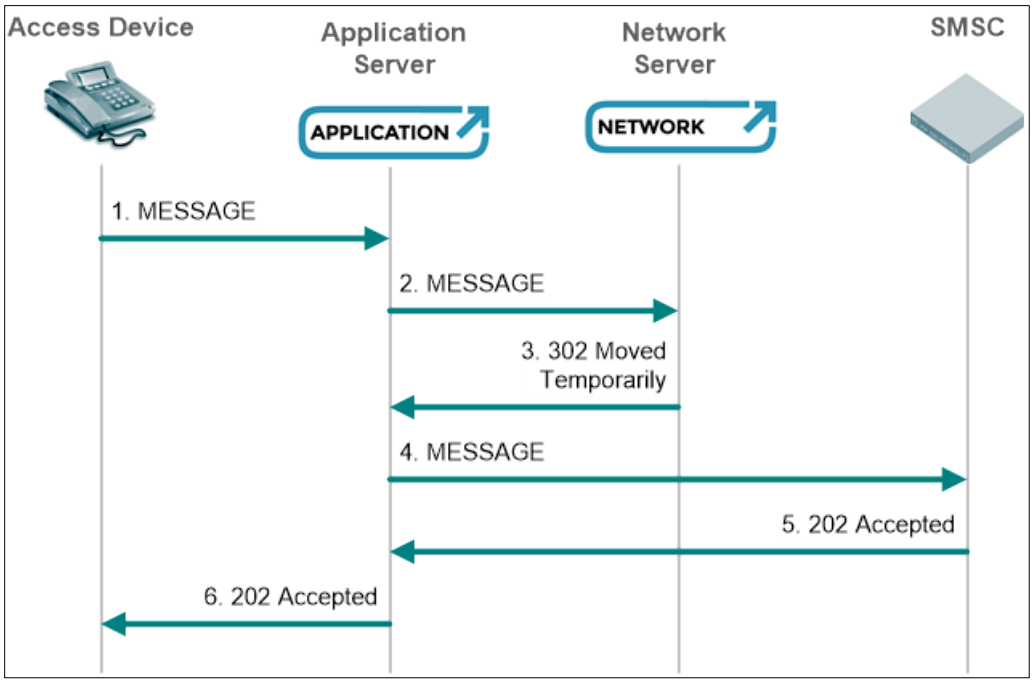

Figure 49 MESSAGE Origination from Cisco BroadWorks Subscriber

©2019 CISCO SYSTEMS, INC.

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

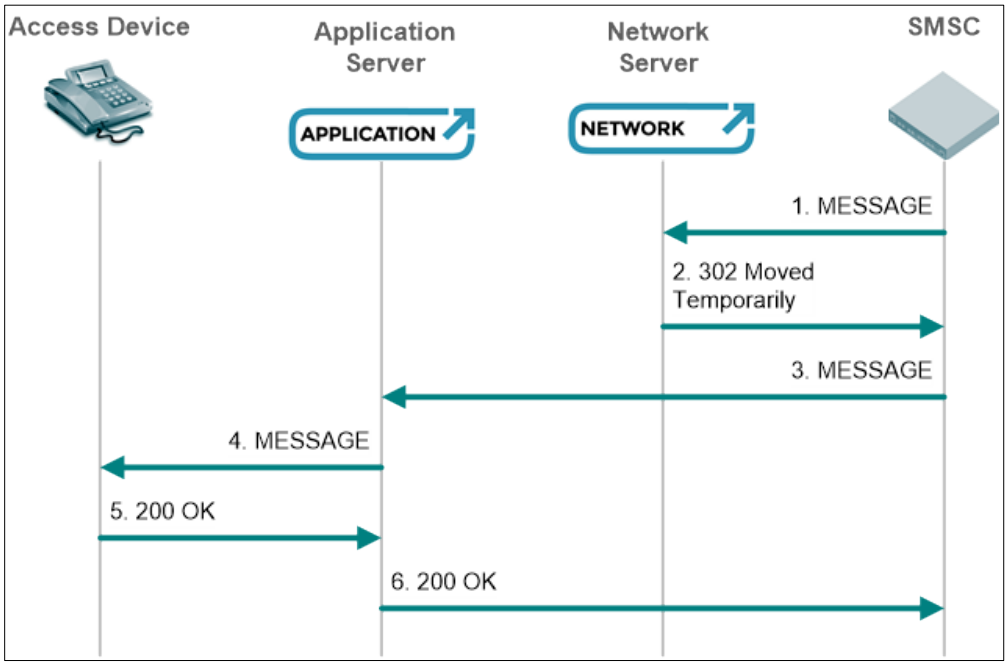

Figure 50 MESSAGE Termination to Cisco BroadWorks Subscriber

NOTE 1: Cisco BroadWorks does not support MESSAGE 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 reflecting the *text/plain* content-type.

```
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!
```


©2019 CISCO SYSTEMS, INC.

3.24 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 DTMF digits for IVR services. If the device connecting to the Media Server supports *RFC 4733* as indicated 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 RTP stream of packets.

Access devices should support *RFC 4733* as it guarantees reliable delivery of DTMF digits from the access device to the entity collecting the digits. Furthermore, compressed codecs such as G.729 are not suitable for sending DTMF tones as waveform data, so that telephone-events must be used whenever a compressed codec is used.

©2019 CISCO SYSTEMS, INC.

3.25 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 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*.

3.25.1 Fax Reception

Incoming fax calls terminate to a dedicated fax number associated with a Cisco BroadWorks subscriber. Cisco BroadWorks supports incoming calls, which include both the audio codec and the fax (udptl) codec in the SDP in the INVITE and calls, which only include the audio codec and negotiate dynamically to T.38.

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

©2019 CISCO SYSTEMS, INC.

3.25.1.1 Message Sequence Diagrams

A successful “faxrecord” session is depicted in the following figure.

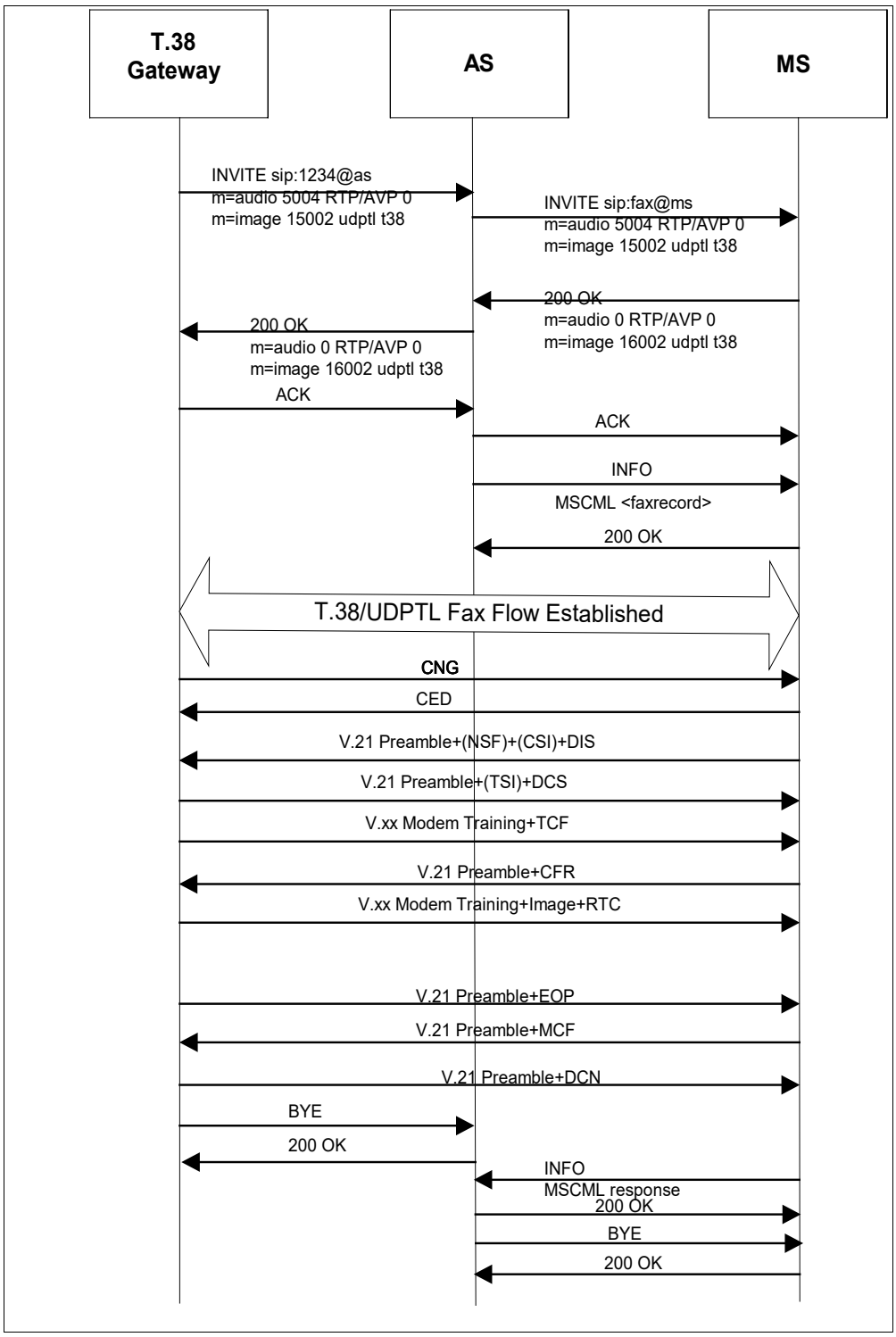

Figure 51 Example of Successful “faxrecord” Session

©2019 CISCO SYSTEMS, INC.

The following figure 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 Media Server re-inviting for a fax session.

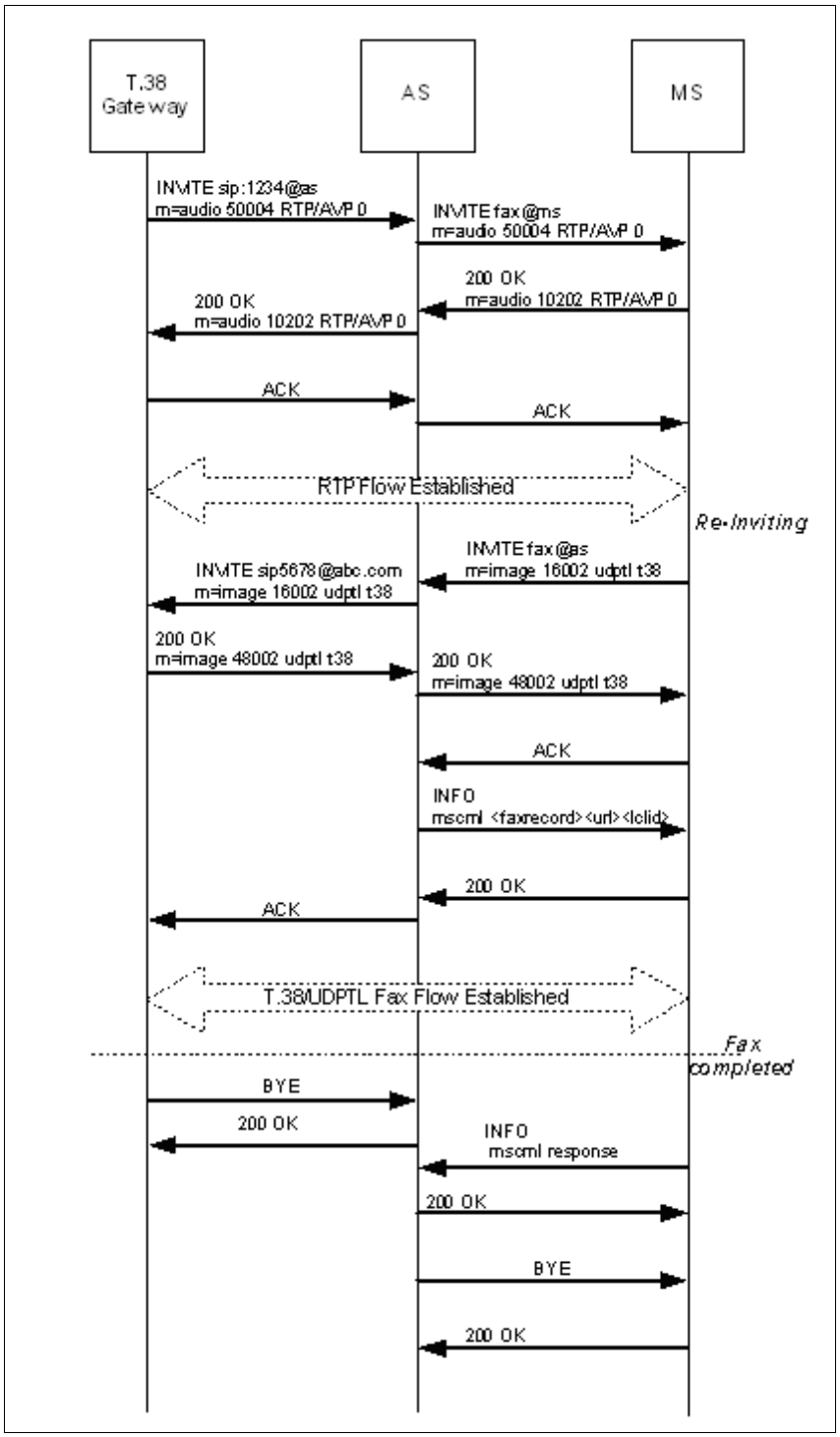

Figure 52 Successful "faxrecord" Call Flow in Re-invite Scenario

©2019 CISCO SYSTEMS, INC.

An unsuccessful “faxrecord” session is depicted in the following figure.

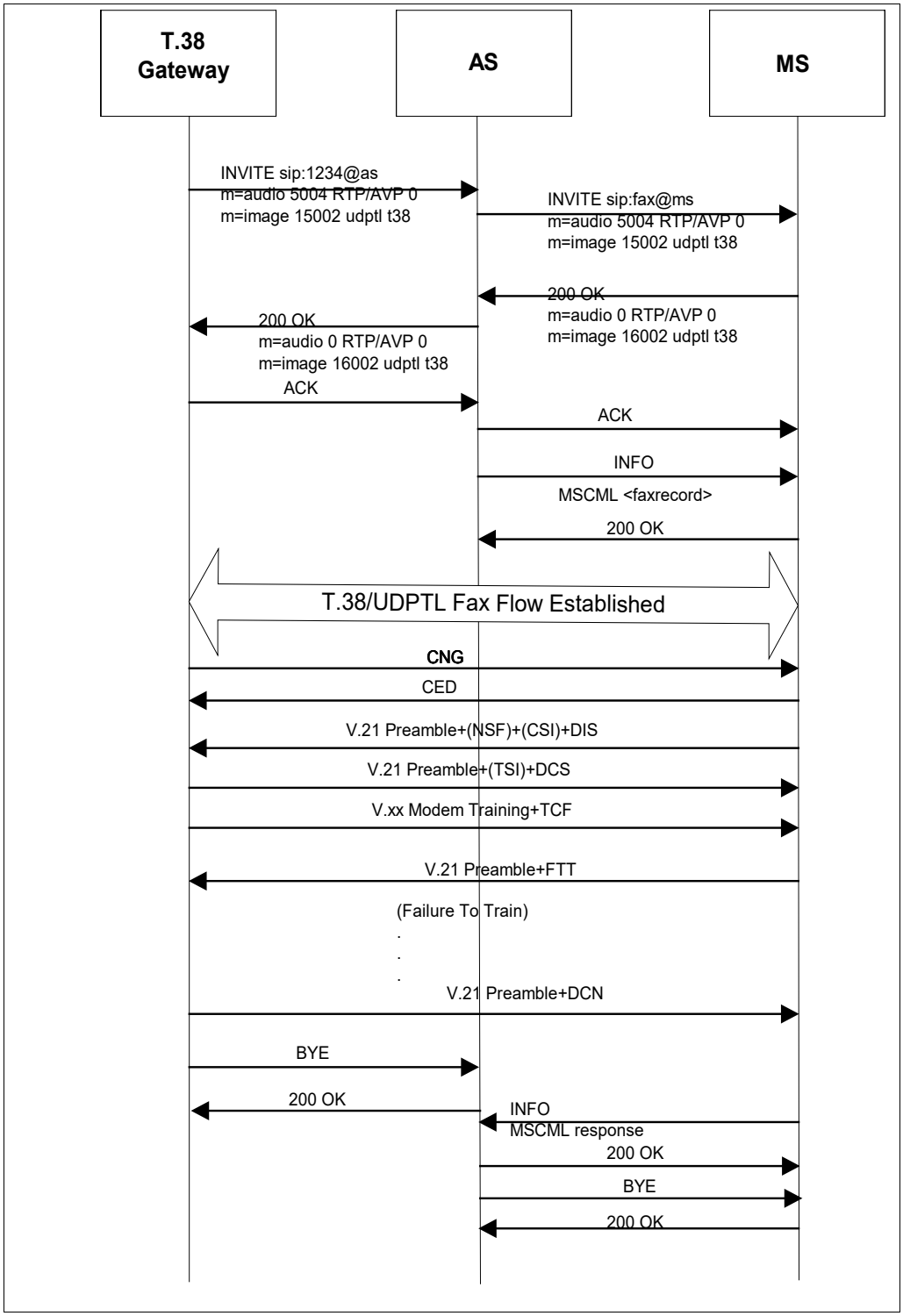

Figure 53 Example of Unsuccessful “faxrecord” Session

©2019 CISCO SYSTEMS, INC.

Once the image data is received, the Media Server stores it in a TIFF file and uses an 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 depicted in the following figure. 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, only one fax header is 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.

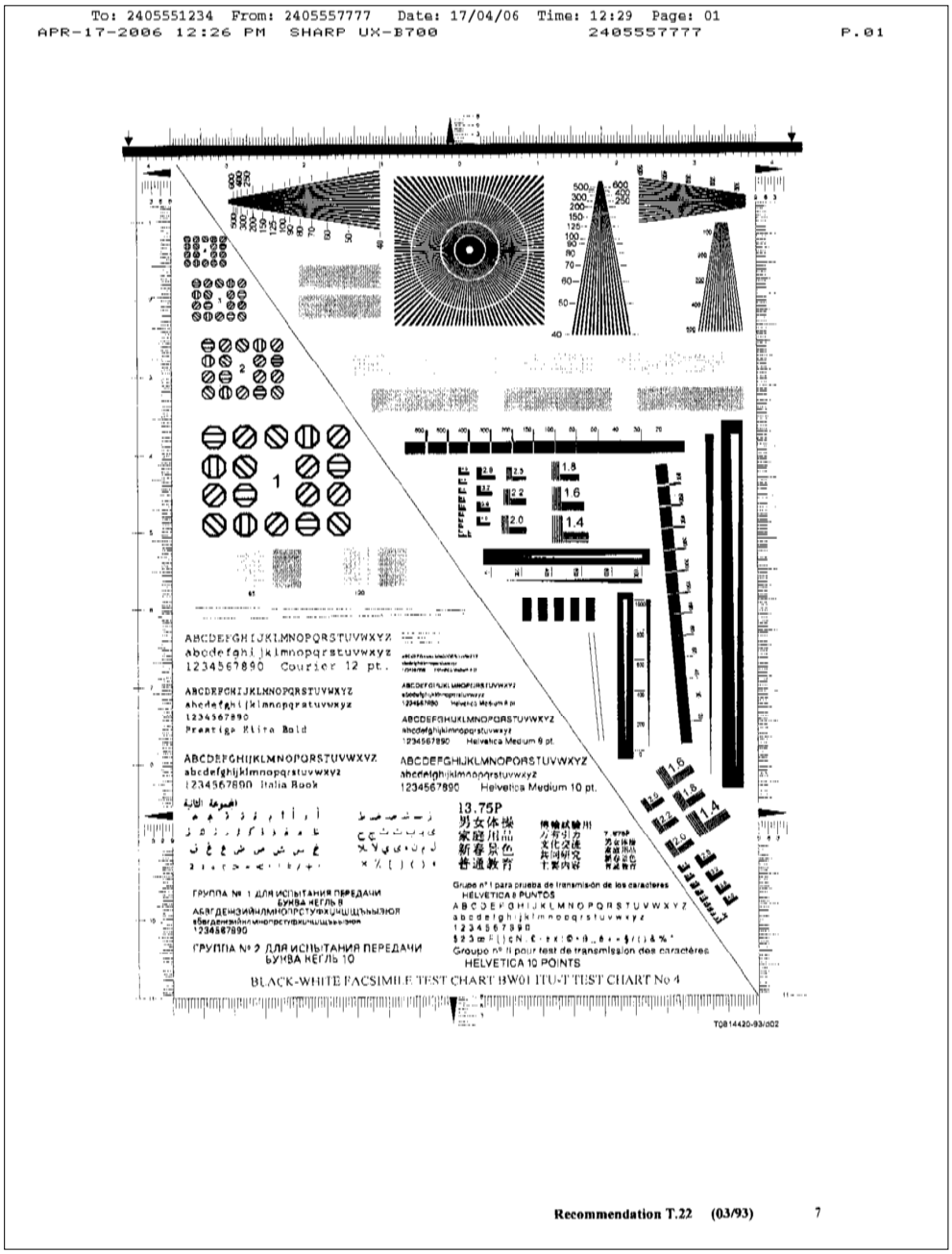

Figure 54 Screen Shot of One Page "faxrecorded" TIFF File with Two Fax Headers

©2019 CISCO SYSTEMS, INC.

3.25.2 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.

3.25.2.1 Message Sequence Diagrams

A successful “faxplay” session setup is depicted in the following figure. 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.

©2019 CISCO SYSTEMS, INC.

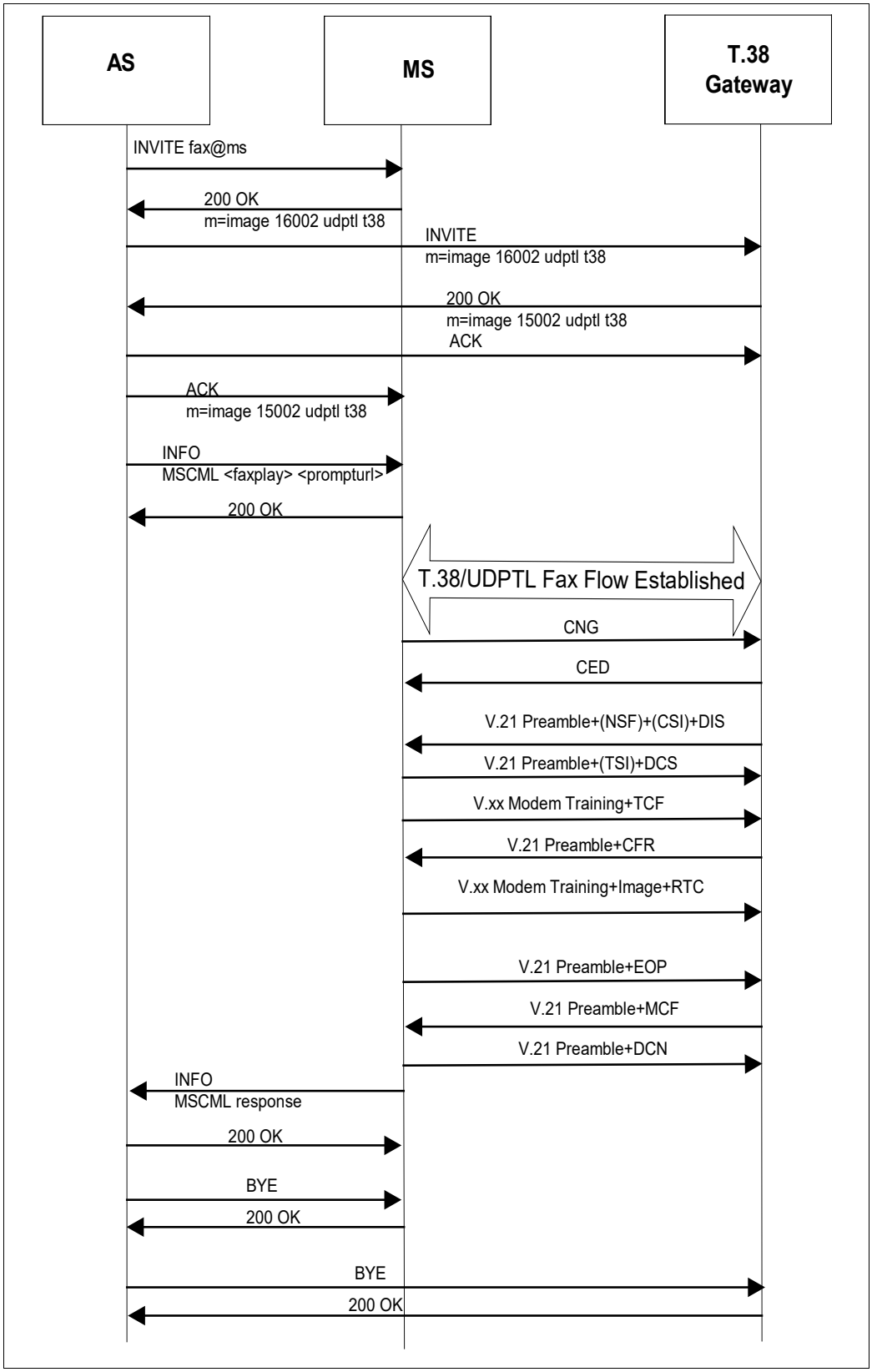

Figure 55 Example of Successful "faxplay" Session

©2019 CISCO SYSTEMS, INC.

A successful “faxplay” session setup (re-INVITE from Terminating T.38 Gateway scenario) is depicted in the following figure.

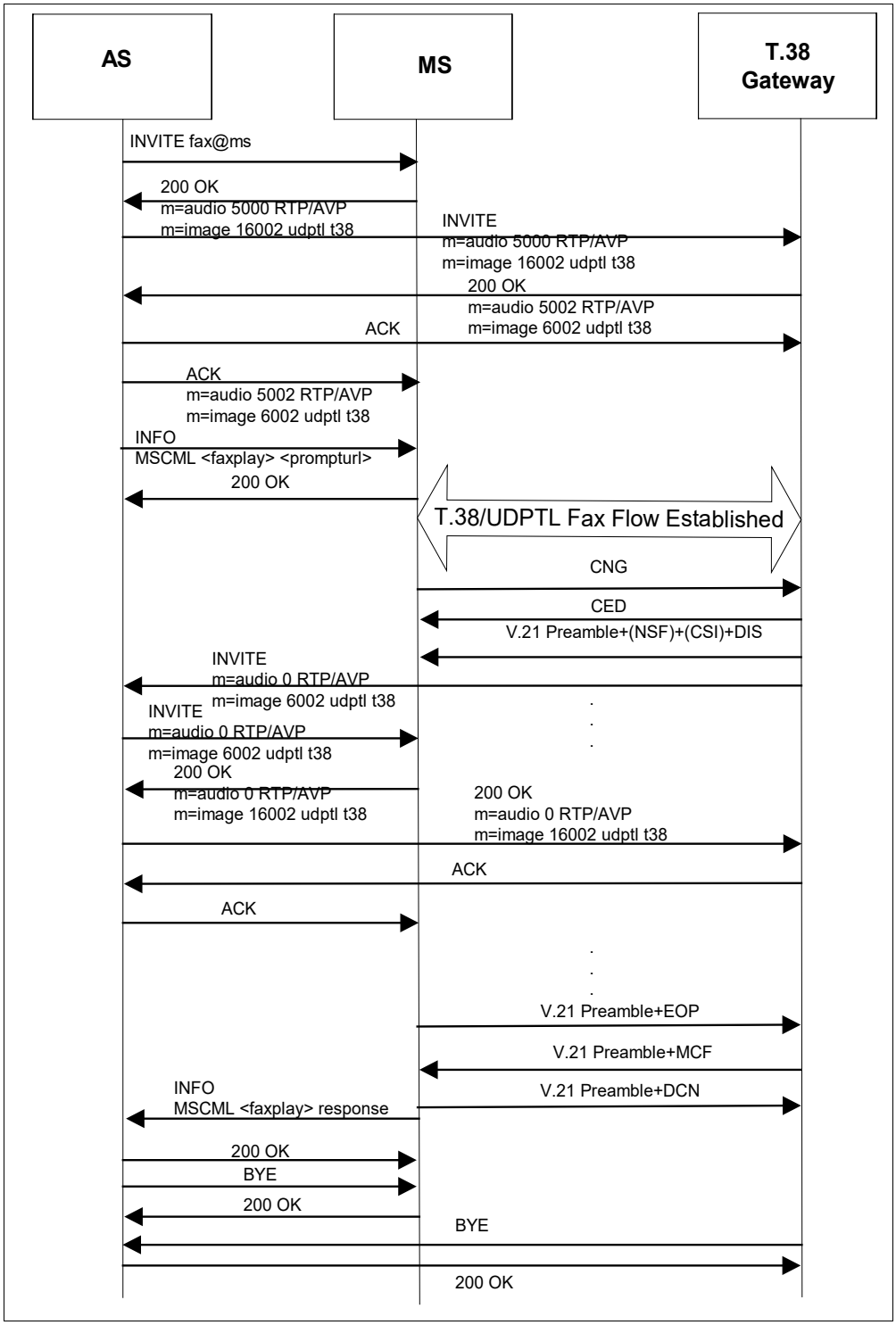

Figure 56 Successful “faxplay” Session in Re-INVITE from Terminating T.38 Gateway Scenario

©2019 CISCO SYSTEMS, INC.

A successful “faxplay” session setup (re-INVITE from Originating T.38 Gateway/Media Server scenario) is depicted in the following figure.

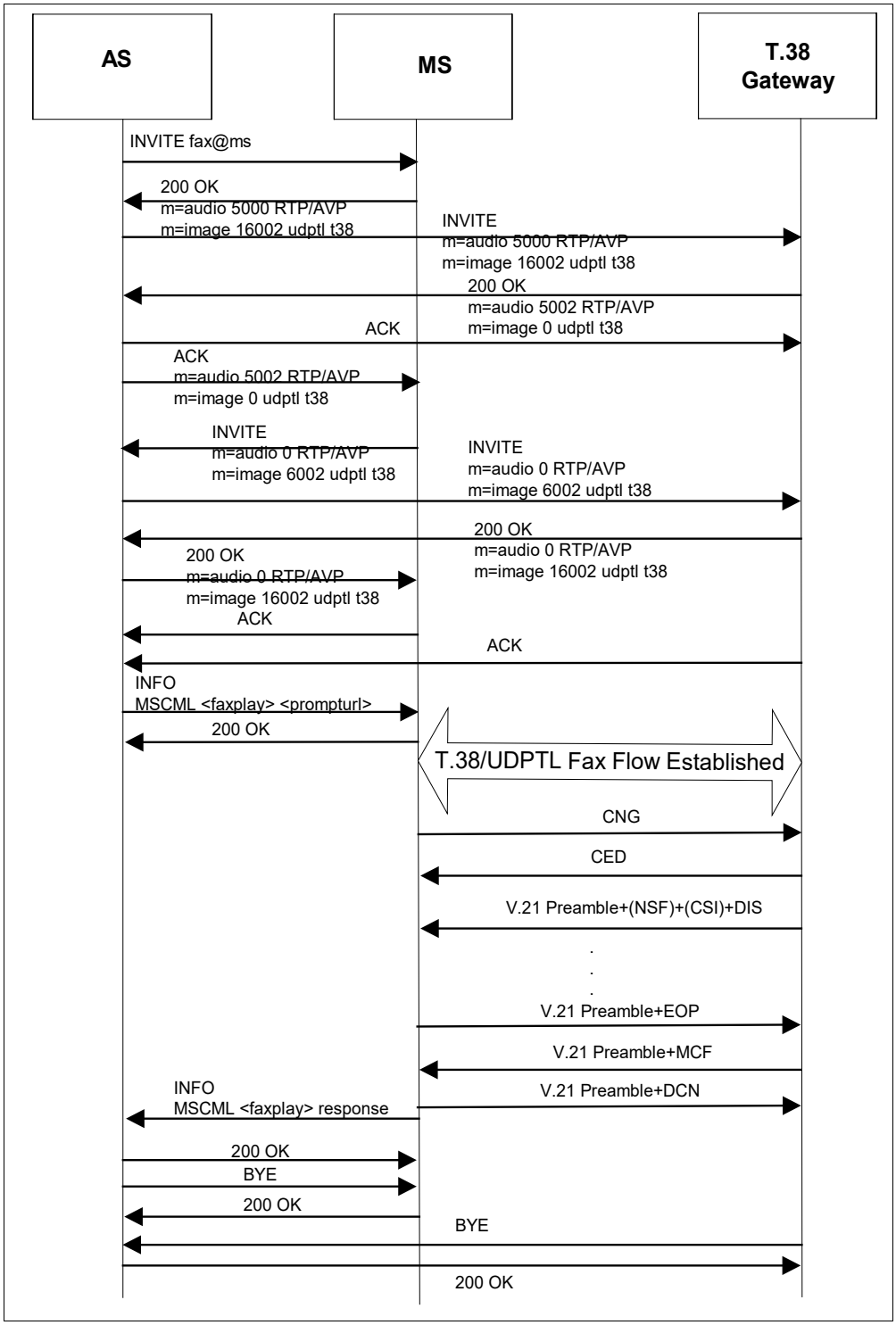

Figure 57 Successful “faxplay” Session in Re-INVITE from Originating T.38 Gateway (Media Server) Scenario

©2019 CISCO SYSTEMS, INC.

An unsuccessful “faxplay” session setup is depicted in the following figure.

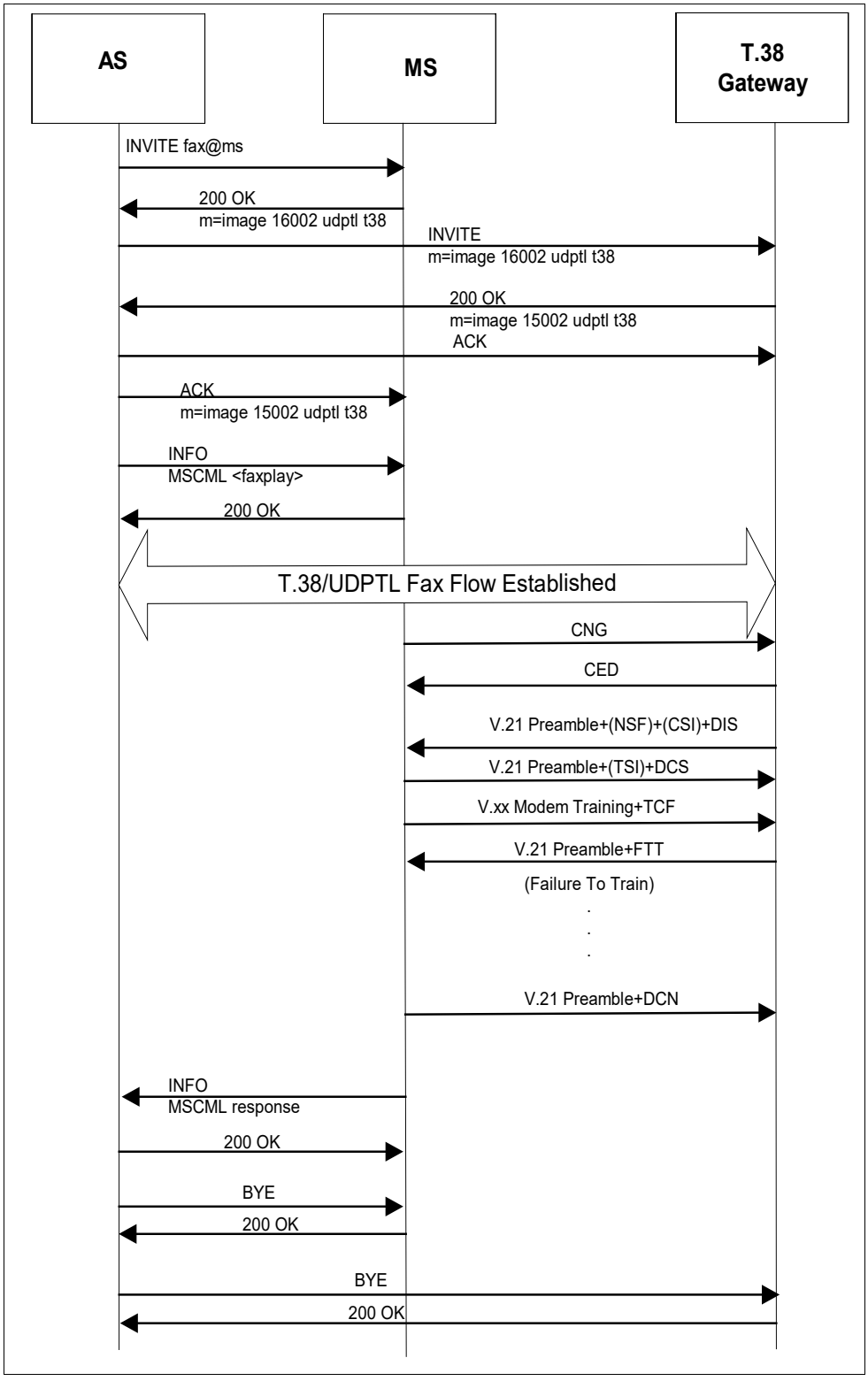

Figure 58 Example of Unsuccessful “faxplay” Session

©2019 CISCO SYSTEMS, INC.

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

The SDP is used to specify media characteristics for a session. An access 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 to *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 "brands" all SDPs that pass through Cisco BroadWorks such that the devices exchanging SDPs view Cisco BroadWorks as the owner of the SDP. As part of this branding, Cisco BroadWorks changes the **o** and **s** lines in the SDP. For example, when a device initiates a request to Cisco BroadWorks with a device-generated SDP, Cisco BroadWorks overwrites the **o** and **s** lines of the SDP from the device, leaving the **m** lines 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 that adhere to the rules of *RFC 3264* and *RFC 4566*, especially for the session id and version contained in the **o** line.

Cisco BroadWorks can interwork devices that use the *RFC 3264* hold mechanism with devices that expect a connection address of 0.0.0.0 to indicate call hold. Cisco BroadWorks converts an SDP with a directionality attribute of *sendonly* or *inactive* to 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 directionality attribute in a received SDP per the 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 and with no directionality attribute, Cisco BroadWorks modifies the SDP to include a directionality attribute of *inactive*. Note that for *sendrecv* SDPs, when the directionality is *sendrecv*, Cisco BroadWorks does not include the directionality 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 an active call with another device provides 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*.

An access device must support the following actions:

- An access device must support receiving a subsequent INVITE (Re-INVITE) with a new session description. The device should swap media streams transparently.

©2019 CISCO SYSTEMS, INC.

- An access device must support receiving an initial INVITE with a session description where the “c” destination addresses 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 backward compatibility. An access 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 access device should include the ***a***=*sendonly*, ***a***=*recvonly*, ***a***=*inactive*, or ***a***=*sendrecv* (default) attribute in the session description it provides, to indicate the disposition of the media it is providing.
- An access device must support receiving a subsequent INVITE (Re-INVITE) with a modified session description where the **c** destination addresses 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 backward compatibility. An access device should support receiving a subsequent INVITE (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 access device should include the ***a***=*sendonly*, ***a***=*recvonly*, ***a***=*inactive*, or ***a***=*sendrecv* (default) attribute in the session description it provides, to indicate the disposition of the media it is providing.
- An access device must support receiving an initial INVITE with no session description (Null SDP). In this scenario, the device should be able to set up a media connection when it receives an ACK with SDP. Additionally, the access device should respond with an SDP in the *200 OK* response to the INVITE, which contains all of the supported codecs of the device in the order of preference. An access device should also support receiving an initial INVITE with a session description with SDP but no media lines. An access device should respond with an SDP in the *200 OK* response to the INVITE with an SDP with no media lines.
- An access device must support receiving a subsequent INVITE (Re-INVITE) with no session description (Null SDP). In this scenario, the device should be able to handle setting up a media connection when it receives an ACK with media. Additionally, the access device should respond with an SDP in the *200 OK* response to the INVITE, which contains all of the supported codecs of the device in the order of preference.
- An access device must support receiving an ACK to an INVITE, which did not have a session description, with a session description where the “c” destination addresses 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 backward compatibility. An access 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 access device should include the ***a***=*sendonly*, ***a***=*recvonly*, ***a***=*inactive*, or ***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, an access 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 “c” destination addresses for the media streams are set to (0.0.0.0) or containing an ***a***=*inactive* attribute, 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 backward compatibility.
  - This scenario can happen when the device is placed “on-hold” via a session description where the “c” destination addresses for the media streams are set to zero (0.0.0.0) or contain an *a=inactive* attribute, and receives a subsequent INVITE (Re-INVITE) with no session description (Null SDP).

©2019 CISCO SYSTEMS, INC.

  - 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 "c" destination addresses 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".

- An access device must support changing media streams in 18x responses and subsequent *200 OK* responses. For calls from an access 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 access 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 access device does not have to support switching from remote ring back to local ring back as Cisco BroadWorks will not initiate this transition. However, it is desirable for the access device to support switch from remote to local ring back.

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

©2019 CISCO SYSTEMS, INC.

3.27 Session Description Protocol Bandwidth Modifiers (RFC 3556)

Cisco BroadWorks may use SDP bandwidth modifiers to prevent an access device from sending RTP media when Cisco BroadWorks provides *Music on Hold*. These bandwidth modifiers are described in *RFC 3556*.

The call flow diagram in *Figure 59* depicts a scenario were Cisco BroadWorks may use the bandwidth modifiers. At the start of this scenario, User A has an established call with User B. User A then puts User B on hold. Cisco BroadWorks identifies the SDP in User A's INVITE request (F1) as a hold SDP and proceeds to provide Music on Hold to User B. Without any action by Cisco BroadWorks toward User A, it is possible that User A could also stream audio media toward User B. To prevent this undesired media from User A, Cisco BroadWorks may add bandwidth modifiers to the answer SDP in the *200* response to User A (F2).

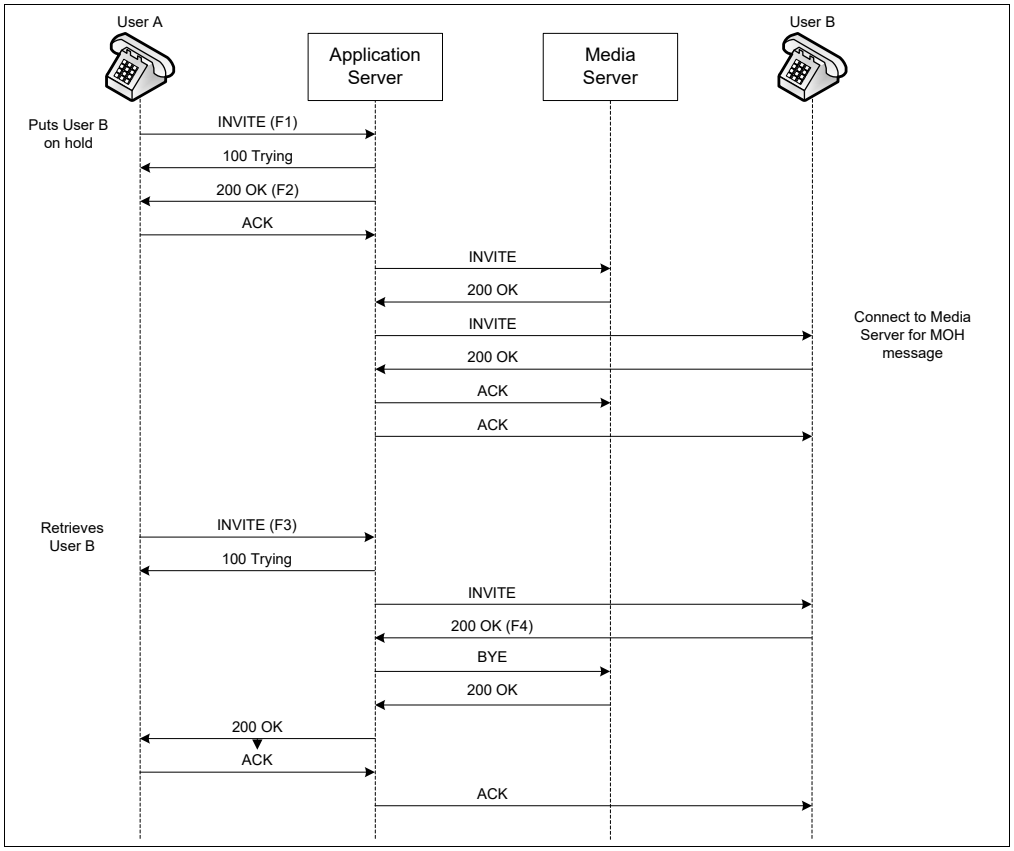

Figure 59 Call Flow Diagram for Music On Hold and SDP Bandwidth Modifiers

The INVITE from User A (F1) might have an offer SDP as follows.

```
v=0
o=user1 53655765 2353687638 IN IP4 127.0.0.1
s=-
c=IN IP4 192.168.8.94
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly
```


©2019 CISCO SYSTEMS, INC.

The answer SDP in the *200* response from Cisco BroadWorks (F2) would then have bandwidth modifiers as follows.

```
v=0
o=BroadWorks 27 2 IN IP4 192.168.8.44
s=-
c=IN IP4 192.168.8.44
t=0 0
m=audio 16426 RTP/AVP 0
b=AS:0
b=RS:800
b=RR:800
a=rtpmap:0 PCMU/8000
a=recvonly
```

The `b=AS:0` line in the SDP sets the RTP bandwidth to 0, which should prevent User A's device from sending RTP media.

Note that Cisco BroadWorks use of bandwidth modifiers in this way is configurable separately for each device. Cisco BroadWorks also supports an alternative method in which it sends a directionality attribute of inactive. Using this alternative method, the answer SDP for the preceding scenario would be as follows.

```
v=0
o=BroadWorks 27 2 IN IP4 192.168.8.44
s=-
c=IN IP4 192.168.8.44
t=0 0
m=audio 16426 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=inactive
```


©2019 CISCO SYSTEMS, INC.

3.28 Cisco BroadWorks Media Type Support

By default, Cisco BroadWorks supports message bodies with the media type *application/sdp*. Cisco BroadWorks supports no other media types by default, but allows any media type to be added as a supported media type via configuration. If Cisco BroadWorks receives an INVITE request that contains a message body with an unsupported media type, Cisco BroadWorks responds with a *415 Unsupported Media* response.

Cisco BroadWorks also supports multipart message bodies when the containing message body has the media type multipart/mixed. Note that it is not necessary to configure support for the multipart/mixed media type. The following rules summarize how Cisco BroadWorks handles message bodies that are part of a multipart message body:

- Cisco BroadWorks checks the media type of each message body. If it recognizes the media type, then it allows that message body and proxies it. Cisco BroadWorks maintains the multipart/mixed body when more than one message body is proxied.
- If Cisco BroadWorks does not recognize the media type, then it silently discards that message body.
- If Cisco BroadWorks discards all the message bodies (because of unsupported media types), then it rejects the request with a 415 Unsupported Media Type response.
- If Cisco BroadWorks discards all the message bodies except one, then it reforms the original multipart message body as a simple (non-multipart) message body.

All media 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 media 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 media types added to the supported content-type list via the Application Server CLI; these media 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 handled 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

©2019 CISCO SYSTEMS, INC.

Method	Content-Type	Proxy to Access Devices and Network	Proxy to Media Server
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 *3.17 SIP INFO Method (RFC 2976)*.

©2019 CISCO SYSTEMS, INC.

3.29 RTP: Transport Protocol for Real-Time Applications (RFC3550)/ 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

Access devices must support codec renegotiation via SIP when using codecs other than G.711 u-law, G.711 a-law, G.726-32, or 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: 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 only can support G.711 u-law. All other devices would negotiate to the codec of choice, as per the rules defined in *RFC 3264*.

©2019 CISCO SYSTEMS, INC.

3.30 Cisco BroadWorks Redundant Application Server Requirements

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

The access device must be able to route to multiple proxies under failure conditions. DNS SRV support is the preferred mechanism for support of redundant proxies. However, redundant proxy support can be accomplished in many ways. Following are some of the possible mechanisms to support this requirement:

- Provide explicit support for a primary and secondary proxy.

  The access device should use the secondary proxy to send messages after receiving Internet Control Message Protocol (ICMP) errors or timing out on the primary proxy. Note that the timeout interval should be relatively short to prevent longer than desirable call setup delays in a failure situation.

- Support FQDN entry for proxy and support DNS getAllHostsByName for resolution of proxy address A records.

  The access device should attempt to route on each A record in the order received by the DNS lookup until a successful route is obtained. The access 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.

- Support FQDN entry for proxy and support DNS SRV records for resolution of proxy address. (Note that this is the preferred approach for redundancy support.)

  The access device should attempt to route on each route in the order received by the DNS SRV lookup until a successful route is obtained. The access device should proceed to the next route after receiving 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.

  Access devices should support a timer mechanism to route advance to the backup proxy. This timer should take precedence over the SIP retransmission rules to ensure timely routing of messages to Cisco BroadWorks in a failure situation.

  Access devices should support the ability to send INVITE requests, REGISTER requests, and other requests to any of the routes, as specified in the above requirements.

  In addition, the access device should use the same route chosen for the proxy for the duration of a given call when possible. Subsequent calls may start with the primary proxy or first route.

  The access device may also support state information as to the last successful route used to communicate with the proxy. In this scenario, the access device would remember the last successful route used to communicate the proxy for a specific duration of time. The access device would continue to use this “cached” route until the “cache” expires. If the access 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.

©2019 CISCO SYSTEMS, INC.

In addition to the static configuration of the device, an access device must support FQDNs in the *Via, Contact, Request-URI, From*, and *To* headers. Additionally, the device must support the following requirements associated with the FQDN in these headers. These requirements ensure that billing records are closed in the case of a failover during an established call, calls are able to complete during a network failure on the access interface, and subsequent requests are handled properly and responded to in a failover situation.

- The access device must support the ability to perform a DNS or DNS SRV lookup on the FQDN in the contact to route subsequent transactions (for example, BYE, and so on).
- The access device must support the ability to perform a DNS or DNS SRV lookup on the FQDN in the *Via* header to route responses if the received path is no longer available.
- The access device must support the ability to detect that the primary route from the DNS or DNS SRV lookup is not available, and then route advance to the next route as specified via the DNS or DNS SRV lookup, for both the *Via* and *Contact* headers.

  The access device must determine that a route is not available in a timely manner and may use ICMP, a timer mechanism, or any other means, to ensure that a route advance occurs in a timely manner during a failure situation.

©2019 CISCO SYSTEMS, INC.

3.31 Cisco BroadWorks Firewall/NAT Traversal Requirements

Cisco BroadWorks also interoperates with NAT/Firewall solutions to allow access devices behind NAT/Firewalls to transparently communicate with Cisco BroadWorks via a Session Border Controller, Application Layer Gateway, or any SIP NAT/Firewall solution. For this solution to work transparently with UDP, certain requirements are placed on the access devices.

The access devices must:

- Support the concept of an Outbound Proxy.
  - The access device must support a configuration option where all messages are sent to an Outbound Proxy, regardless of the request-URI in the outbound request.
  - The Outbound Proxy redundancy requirements are identical to the Proxy requirements described in the previous section. The access device must support at least one of the redundancy requirements for the Outbound Proxy functionality.
  - The access device must populate the *From* header with the address of Cisco BroadWorks (that is, proxy) for the host portion of the SIP URI.
- Support Symmetric UDP for SIP signaling.
  - The access device must support sending/receiving SIP messages on the same port. This is required to keep the NAT binding open on the firewall.
  - For example, if the device accepts SIP messages on port 5060, then it must send SIP messages using port 5060 such that the source IP port is 5060.
- Support Symmetric UDP for RTP sessions.
  - The access device must support sending/receiving RTP media on the same port. This is required to keep the NAT binding open on the firewall.
  - For example, if the device accepts RTP messages on port 10020, then it must send RTP messages using port 10020 such that the source IP port is 10020.
  - The access device must support receiving unsolicited NOTIFYs for an unrecognized event package. Some NAT/Firewall traversal solutions use the unsolicited NOTIFY to keep the NAT binding open on the firewall for the SIP signaling messages. The RTP sessions do not require such a keep alive, as the RTP packets occur often enough during the session to keep the NAT binding for the RTP port open. Note that Voice Activity Detection (VAD) must be disabled for devices behind a NAT, for the NAT binding to remain open.
  - The access device must provide a response to the NOTIFY. The access device does not have to provide a *200 OK* response, but it must respond with a final response.

©2019 CISCO SYSTEMS, INC.

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 *3.30 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 *RFC 3261, section 4.3*. By following these RFC procedures, the device will effectively contact the other Application Server during overload conditions.

When the Application Server is overloaded and returns a *503 Service Unavailable* response, it may 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.

©2019 CISCO SYSTEMS, INC.

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 service unavailable time period is set. During this service unavailable time period, the Application Server throttles the SIP requests sent to the overloaded neighbor.

©2019 CISCO SYSTEMS, INC.

3.33 Cisco BroadWorks Access Device Configuration Requirements

Cisco BroadWorks provides the ability to integrate Cisco BroadWorks subscriber provisioning with access device configuration file management. Cisco BroadWorks automatically populates subscriber information into specific access device configuration file formats.

To provide this capability, Cisco BroadWorks supports the following access device configuration file management characteristics. An access device must support these characteristics to make use of this functionality with Cisco BroadWorks.

- TFTP/FTP file access for access device configuration files
- Access device configuration files in nested format or system/individual format
- Provisioning support of the following type of fields in the access device configuration files, shown in the following table.

TAG	Definition	Notes
%BWCLID-#%	CLID (First and last names).	This is the subscriber's calling name used for Calling Line Identity.
%BWEXTENSION-#%	Subscriber's extension is retrieved from the extension provisioned in the user's profile. If an extension has not been provisioned, the tag is replaced with the subscriber's phone number (DN).	
%BWCLID-#%	CLID (First and last names).	This is the subscriber's calling name used for Calling Line Identity.
%BWLINEPORT-#%	Line port provisioned for the subscriber.	This is the access device address of record.
%BWHOST-#%	Domain portion of the provisioned line port for the device assigned to the user. Retrieved from the user's profile.	When the BWHOST tag is used, the domain for the assigned device must be routable via the DNS server.
%BWLINEPORTHOST-#%	Line port and domain provisioned on the device that is assigned to the user. The line port and domain are concatenated together to create a valid address (for example, linePort@domain). The @ symbol is included. Retrieved from the user's profile.	
%BWNAME-#%	Subscriber's first and last names.	
%BWLASTNAME-#%	Subscriber's last name.	
%BWFIRSTNAME-#%	Subscribers' first name.	

©2019 CISCO SYSTEMS, INC.

TAG	Definition	Notes
%BWSHAREDLINE-#%	Indicates if a line is shared or private. If the subscriber has been assigned Shared Call Appearance, this tag is used in the configuration file to indicate that the line is shared. Only the lines provisioned on the Shared Call Appearance page are set to "Shared" in the configuration file. If Shared Call Appearance has not been assigned, the line is considered private.	This indicator is used for Enhanced Shared Call Appearance.
%BWAUTHUSER-#%	Authentication user name.	
%BWAUTHPASSWORD-#%	Authentication password.	
%BWSOFTWARELOAD%	Device software load.	Not supported.
%BWDEVICEID%	Provisioned device ID.	Typically used as a comment in the configuration template file.
%BWTIMESTAMP%	Time the file was built.	Some access devices use the time stamp as the Sync value for remote reboot support. This can also be used as a comment in the template file.
%BWDISPLAYNAMECLID%	CLID of the first private line.	Can be used as the display name on the phone.
%BWDISPLAYNAMELINEPORT%	Line or port of the first private line.	Can be used as the display name on the phone.
%BWDISPLAYNAMEDN%	DN of the first private line.	Can be used as the display name on the phone.
%BWSERVERADDRESS%	Access cluster host name or if the cluster host name is not found, the value is the private IP address.	Used as the proxy server address in the template file. If the server address cannot be found, the value is set to "ENTER SIP SERVER IP ADDRESS or HOST NAME".
%BWFILESERVERLOCATION%	Location of the phone's file server.	
%BWFILESERVERDIR%	Directory where the configuration files are stored on the FTP server.	Typically this should not be used for access devices that only support Trivial File Transfer Protocol (TFTP) configuration file access.

Additionally, Cisco BroadWorks also supports the ability to remotely reboot an access device. The remote reboot capability is typically used after an access device configuration file change, to force the device to upload the updated configuration file. This allows a service provider to upgrade access devices without requiring end-user intervention or direct customer premises equipment (CPE) access.

©2019 CISCO SYSTEMS, INC.

Cisco BroadWorks sends a NOTIFY method with a *check-sync Event* header to remotely reboot the device. Cisco BroadWorks fully supports authentication of this NOTIFY method.

The access device must not reboot the phone while there is an active call. In addition, the access device must independently manage the receipt of NOTIFY methods with the *check-sync Event* header and active calls on the device.

©2019 CISCO SYSTEMS, INC.

3.34 Cisco BroadWorks Video Device Requirements

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

3.34.1 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.

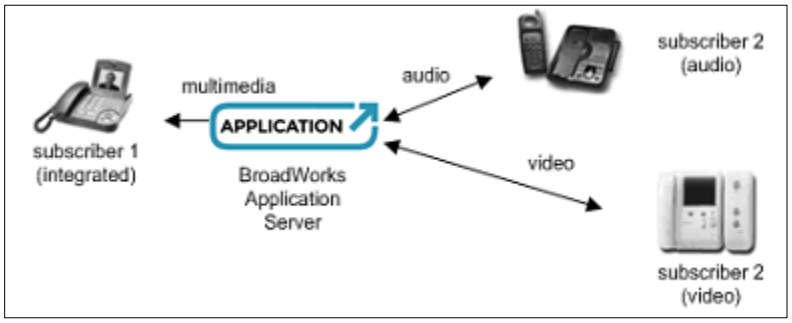

Figure 60 Cisco BroadWorks Video Add-On Service

- **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:

©2019 CISCO SYSTEMS, INC.

- 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 the ***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 an ***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 of Call-Info auto-answer parameter described in the *BroadWorks SIP Access Side Extensions Interface Specification* [43].
  - Client configuration for auto-answer; this is a local policy on the device to auto-answer all incoming calls.

3.34.2 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:

- Plays synchronized audio and video streams over RTP.
- Records synchronized audio and video streams over RTP.
- Plays synchronized audio and video streams over RTP, while performing DTMF digit collection simultaneously.
- Plays the audio portion of an audio + video file.
- Records a synchronized audio and video stream, while a separate audio and video stream is played.

3.34.2.1 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.

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

3.34.2.2 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.

©2019 CISCO SYSTEMS, INC.

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 also 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 per second and a maximum bit stream of 64 KB per second. Level 20 represents a resolution up to 352 x 288 pixels at 30 frames per second and a maximum bit stream of 128 KB per second.

3.34.2.3 Cisco BroadWorks Video Device Requirements for Video IVR

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

- Support of *RFC 4629*
- Support of *RFC 5168* (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

3.34.2.4 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 (with a 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 described above)
    - QCIF with MPI determined (as described above)

©2019 CISCO SYSTEMS, INC.

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

- If profile/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 baseline profile results in rejection of the RTP payload.
- If profile/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 are 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 is 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 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
```


©2019 CISCO SYSTEMS, INC.

```
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
```


©2019 CISCO SYSTEMS, INC.

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 are 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, it 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 the maximum picture size of QCIF with MPI of 2 or higher.
- Level 20 supports CIF with MPI of 2 or higher and QCIF and smaller with MPI of 1 or higher.
- Levels 30 and 40 support CIF MPI of 1 or higher.
- Levels 50 and 60, when you eliminate picture sizes > CIF, supports CIF at fps <= 50 and 352x240 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.

3.34.2.5 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
```


©2019 CISCO SYSTEMS, INC.

```
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
```

In addition, when the Media Server generates an offer SDP and the 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
```

3.34.2.6 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 ignores these options until industry consensus is established in that area.

3.34.2.7 H.264 Parameter Sets Over RTP

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 SIP/SDP offer/answer signaling and/or over the 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.

©2019 CISCO SYSTEMS, INC.

3.35 Deny Calls From Unregistered Users

Cisco BroadWorks supports the ability to block originating calls when the originating device has not registered. This functionality provides a basic access control list that can be dynamically maintained via SIP registration.

At a high level, this origination screening works as follows. When the Application Server receives a new INVITE request on the access interface, it creates a list of source addresses collected from the request's *Via* header fields. At the same time, it also creates a list of device addresses collected from SIP registration and provisioned addresses. Before allowing the call to proceed, the Application Server checks to see if one of the source addresses matches one of the device addresses. If it finds a match, it allows the call to continue; if not, it blocks the call with a *Forbidden* treatment. (By default, the *Forbidden* treatment causes the Application Server to send a SIP *403* response.)

To enable this screening, the system parameter *denyCallsFromUnregisteredUsers* must be set to "true". This parameter is accessible from the CLI level *AS_CLI/System/Registration*.

Emergency calls and repair calls are always exempt from this screening.

The following points provide details of the procedure the Application Server uses to create the source address list:

- If a *Via* entry has a *received* parameter, then the Application Server adds the value of that parameter to the source address list and ignores the address in the *sent-by* field.
- If a *Via* entry does not have a *received* parameter, then the Application Server adds the value of the *sent-by* field to the source address list, provided it is an IPv4 address or an IPv6 address. If the *sent-by* field contains an FQDN, then the Application Server omits it from the list.
- The Application Server applies special treatment to the most recent *Via* header entry. If the *sent-by* field in the entry does not match the actual source address, the Application Server proceeds as though that *Via* entry had a *received* parameter with the actual source address. Consequently, the Application Server adds the actual source address to the source address list and ignores the address in the *sent-by* field.
- The source address list may contain both IPv4 addresses and IPv6 addresses.

The following points provide details of the procedure the Application Server uses to create the device address list:

- The Application Server collects device addresses from three sources:
  - Any contact URI from an unexpired SIP registration for the device endpoint.
  - Any contact URI provisioned for the device endpoint. (Such a contact is sometimes called a "static registration".)
  - A device address provisioned for the access device.
- If a contact URI has a *maddr* parameter, then the Application Server adds the value of that parameter to the device address list and ignores the rest of the URI.
- If a contact URI does not have a *maddr* parameter, then the Application Server takes the device address from the host part of the URI.
- If any of the addresses is an FQDN, the Application Server performs DNS queries to resolve the FQDN to zero or more IP addresses. Depending on the system configuration, the Application Server may perform queries for NAPTR, SRV, A, and AAAA records.

©2019 CISCO SYSTEMS, INC.

- The device address list may contain IPv4 and IPv6 addresses.

When the Application Server attempts to match a source address to a device address, it considers only the IP address. This means it does not consider the transport (UDP, TCP) or the transport port number when attempting the match.

©2019 CISCO SYSTEMS, INC.

3.36 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

3.36.1 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.

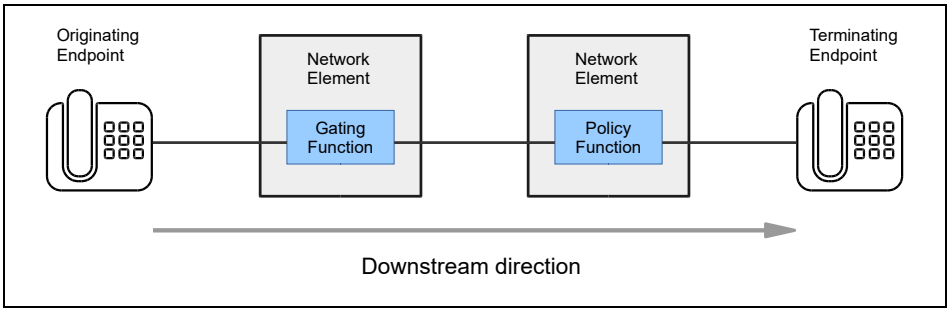

Figure 61 Gating Function and Policy Function

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[9] 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.

[9] 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.

©2019 CISCO SYSTEMS, INC.

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.

©2019 CISCO SYSTEMS, INC.

- 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".

3.36.2 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 *3.10 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.

©2019 CISCO SYSTEMS, INC.

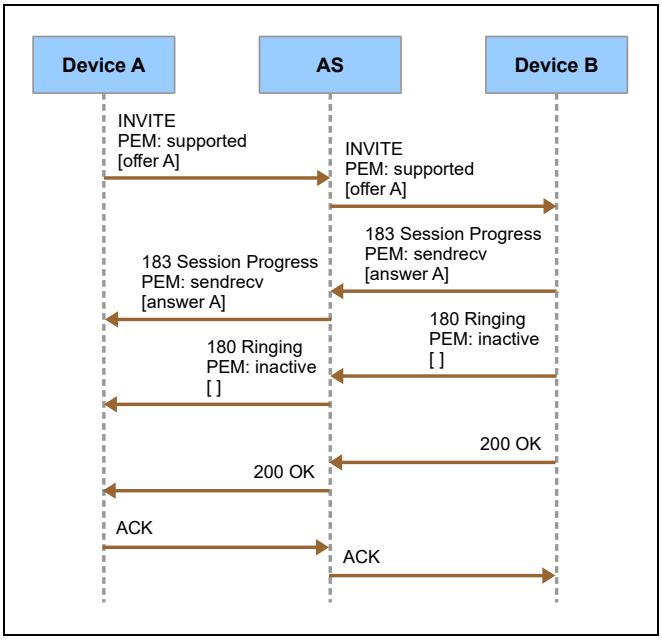

Figure 62 Early Media Transition with P-Early-Media and Gating Function Ringback

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.

©2019 CISCO SYSTEMS, INC.

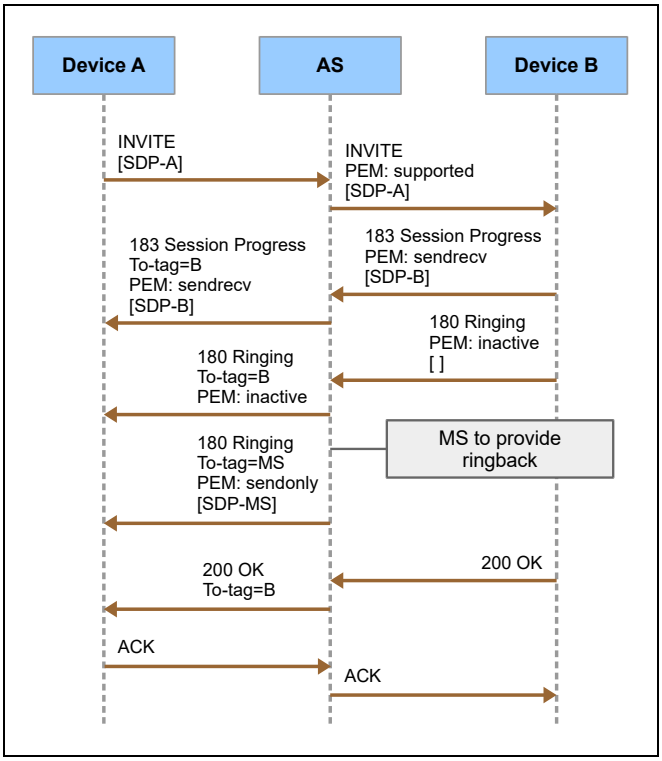

Figure 63 Early Media Transition with P-Early-Media and Media Server ringback

3.36.3 Interactions With SIP Forking

3.36.3.1 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.

©2019 CISCO SYSTEMS, INC.

3.36.3.2 Cisco BroadWorks Forking Services

Cisco BroadWorks supports many forking services, such as Shared Call Appearance, Cisco 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 *3.9 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.

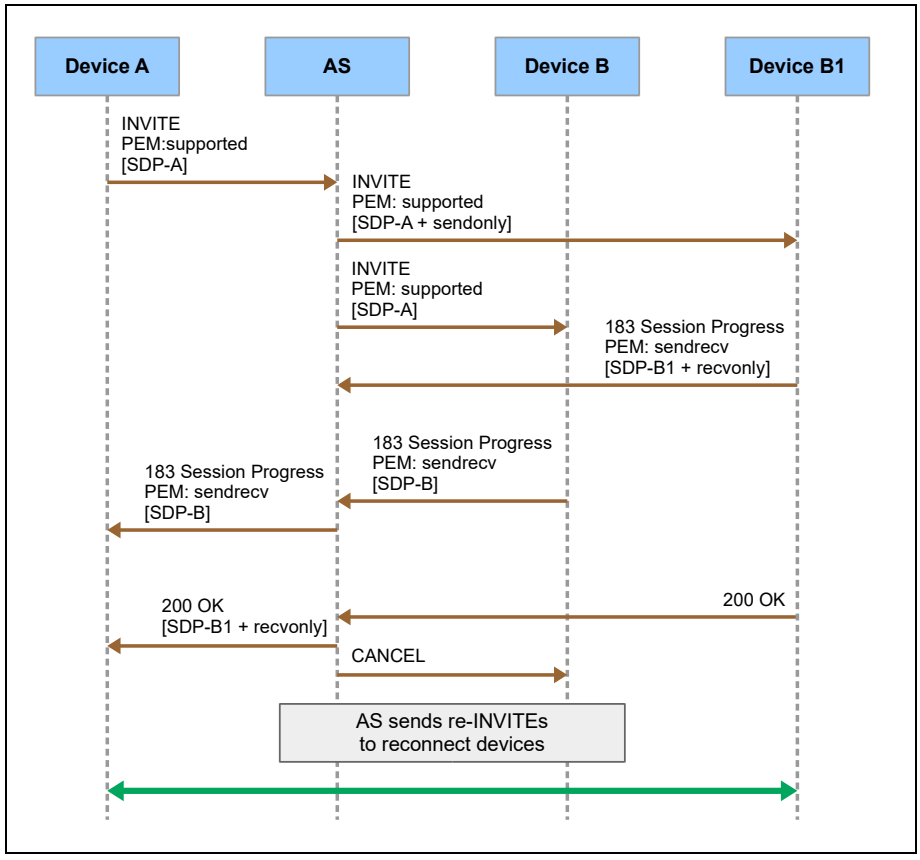

Figure 64 Shared Call Appearance with Application Server Consuming Provisional Responses from Secondary Endpoint

©2019 CISCO SYSTEMS, INC.

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". See section *3.9 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".

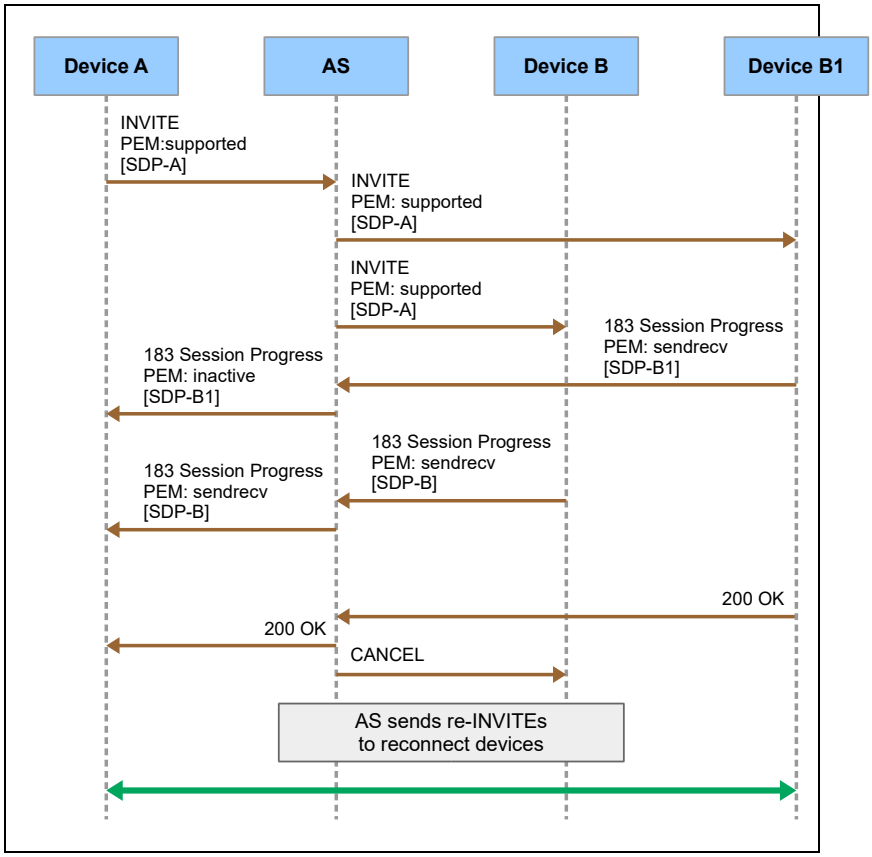

Figure 65 Shared Call Appearance with Application Server Relaying Provisional Responses from Secondary Endpoint

©2019 CISCO SYSTEMS, INC.

3.36.3.3 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 is an existing dialog.

The following simplified call flow diagram depicts 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.

©2019 CISCO SYSTEMS, INC.

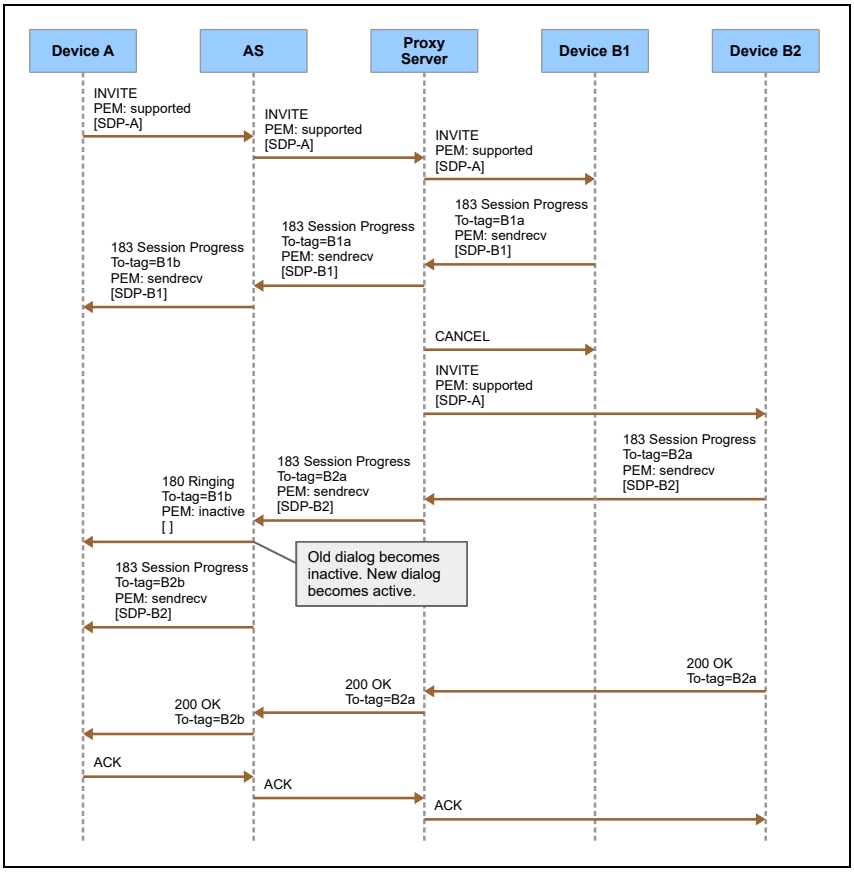

Figure 66 Proxy Server Forking and Early Media Source Selection

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.

©2019 CISCO SYSTEMS, INC.

The following simplified call flow diagram depicts 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.)

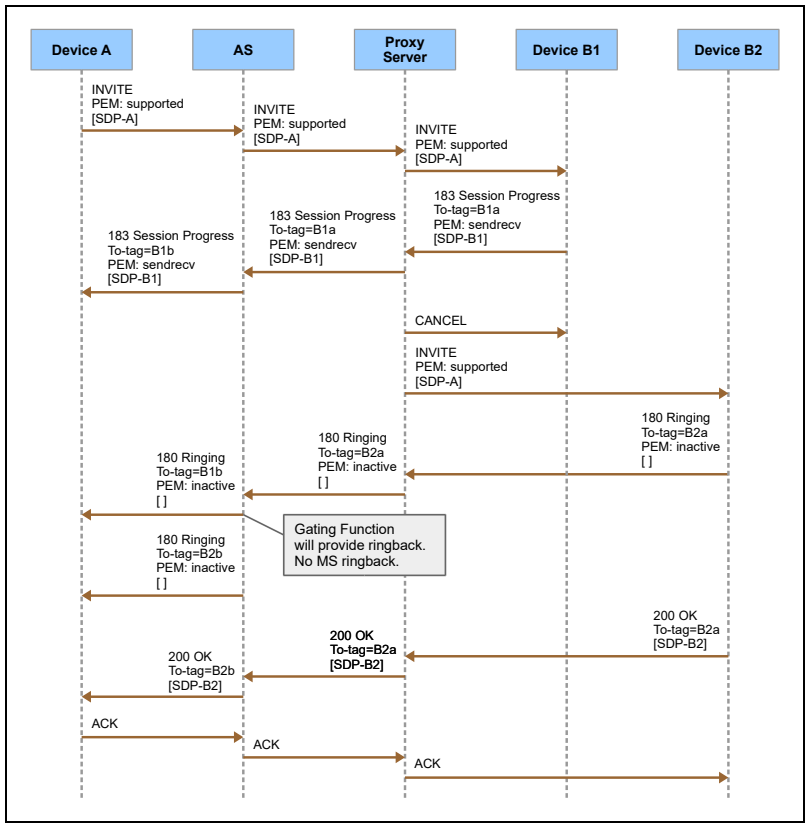

Figure 67 Proxy Server Forking with Gating Function Ringback

The following simplified call flow diagram depicts 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.

©2019 CISCO SYSTEMS, INC.

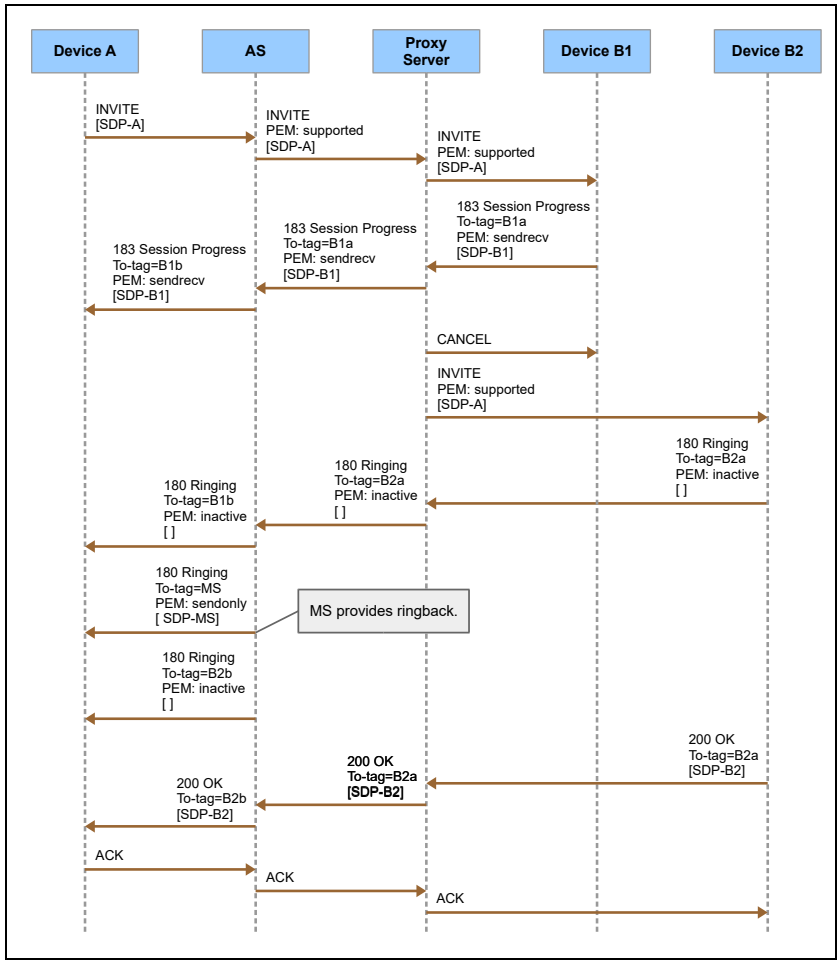

Figure 68 Proxy Server Forking with Media Server Ringback

©2019 CISCO SYSTEMS, INC.

3.37 Configurable Treatments and Reason Header (RFC 3326)

3.37.1 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* [45]), 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 *BroadWorks Treatment Guide* [46]. 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

  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.

©2019 CISCO SYSTEMS, INC.

- 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.

3.37.2 Reason Header

3.37.2.1 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" / token
reason-params = protocol-cause / reason-text
              / reason-extension
protocol-cause = "cause" EQUAL cause
cause = 1*DIGIT
reason-text = "text" 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.

3.37.2.2 SIP Protocol

Cisco BroadWorks supports the “SIP” protocol in the *Reason* header via configurable treatments.

Independently of the treatments configuration, Cisco BroadWorks adds a *Reason* header with the “SIP” protocol in a CANCEL request to a device endpoint when the call is answered at a different endpoint. See section *3.37.3 Forking Services*.

3.37.2.3 Q.850 Protocol

Cisco BroadWorks supports the “Q.850” protocol in the *Reason* header via configurable treatments.

3.37.2.4 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 in the following list:

- “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
```


©2019 CISCO SYSTEMS, INC.

```
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", "xfer-cc-front" – Cisco BroadWorks does not send either of these parameters to an access device.
- "bw-internal-forbidden" – Cisco BroadWorks does not send this parameter to an access device.

3.37.2.5 BW-NS Protocol

Cisco BroadWorks uses the "BW-NS" *Reason* protocol only in SIP messages exchanged between the Application Server and the Network Server. Cisco BroadWorks does not use this protocol on the access interface.

3.37.2.6 Diversion Protocol

When the 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
```


©2019 CISCO SYSTEMS, INC.

```
Reason:Diversion;text="busy"
```

3.37.3 Forking Services

Cisco BroadWorks supports many forking services, such as Shared Call Appearance, Simultaneous Ring, and more. For these services, Cisco BroadWorks sends simultaneous INVITE requests to multiple locations. After one location answers, Cisco BroadWorks sends CANCEL requests to the other locations. To provide more precise information to the non-answering locations, Cisco BroadWorks adds a *Reason* header to the CANCEL request with the reason text “Call completed elsewhere”. The following is an example of this CANCEL request.

```
CANCEL sip:102@10.16.145.5 SIP/2.0
Via:SIP/2.0/UDP 10.16.145.3;branch=z9hG4bKBroadWorks.16smkct-
10.16.145.5V5060-0-146434168-1571676787-1374682403054-
From:"John Q.
Public"<sip:2145550000@initech.test;user=phone>;tag=1571676787-
1374682403054-
To:"Dom Portwood"<sip:102@initech.test>
Call-ID:BW111323054240713-189664729@10.16.145.3
CSeq:146434168 CANCEL
Reason:SIP;text="Call completed elsewhere";cause=200
Max-Forwards:10
Content-Length:0
```


©2019 CISCO SYSTEMS, INC.

3.38 Registration

SIP phones may be configured on Cisco BroadWorks with static or dynamic contacts, depending on device type.

Static contacts are manually configured on the web interface and specify the location where the device can be contacted.

Dynamic contacts are obtained when the access device registers to Cisco BroadWorks using the REGISTER request, as defined in *RFC 3261*.

Cisco BroadWorks supports GIN registration, as described in *RFC 6140*.

- Cisco BroadWorks accepts a *Require* header with the value *gin* in a REGISTER request from an access device.
- Cisco BroadWorks accepts a Contact URI with a *bnc* parameter in a REGISTER request from an access device.
- Cisco BroadWorks uses an implicitly generated location database entry to send an INVITE request to an access device, as required by GIN registration.
- The implicitly generated location database entries depend on the configuration of a Trunk Group in Cisco BroadWorks.

For more information on GIN registration, see the *BroadWorks SIP Trunking Solution Guide* [66].

3.38.1 Network Server Redirection for REGISTER

The Network Server can route a SIP REGISTER request coming from the access side to the Application Server configured to process the SIP request. When the Network Server is configured with this routing capability, access devices and session border controllers (SBCs) do not require any provisioning to map all the endpoints to their respective Application Server in the network. In this model, access devices point to the SBCs and SBCs route access device SIP messages to the Network Server. Based on the SIP message content, the Network Server finds the Application Server on which the originating user is hosted. The Network Server then sends back to the SBC a SIP *302* response containing an ordered list of Application Servers to be contacted. The SBC must redirect its request to the destination(s) specified in the *Contact* header of the *302* response. The request then proceeds as usual to the appropriate Application Server.

Figure 69 describes the SIP messaging flow for a REGISTER using the Network Server as the Application Server finder.

©2019 CISCO SYSTEMS, INC.

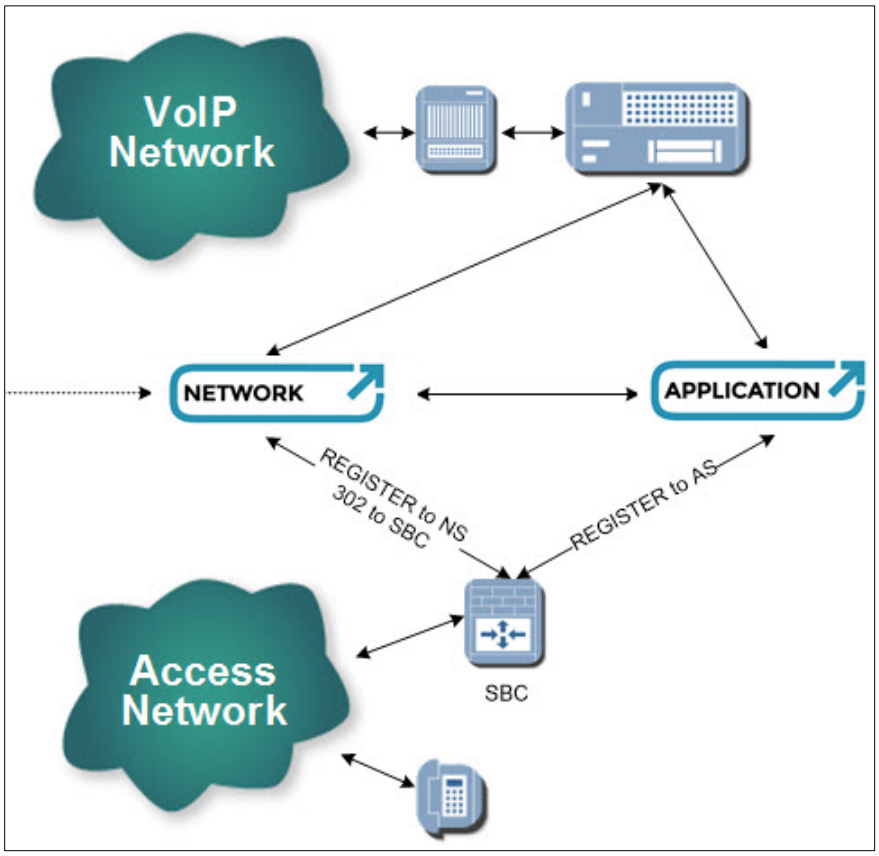

Figure 69 SIP Messaging Flow for Call Setup

To route an access side request, the Application Server (optionally) synchronizes with the Network Server the line/port it is serving.

When the line/port values are synchronized, the Network Server can select the appropriate Application Server cluster based on the line/port value taken from the SIP request.

When the line/port values are not synchronized, some restrictions apply to the selection of the line/port values to enable the Network Server to appropriately identify the serving Application Server cluster.

- The line/port of the user must be its DN in E164 format, or
- The line/port of the user must be its DN in E164 format less the country code, or
- The line/port must be a number that is converted to the user, which must be its DN in E164 format being processed by a digit manipulation operation configured on the Network Server. A different digit manipulation operation can be associated for each range of line/port number.
- The domain portion of the caller URI must not correspond to any existing network element known by the Network Server. Otherwise, if the combination DN and domain corresponds to an existing user, the profile associated with that user is used and the call may be routed differently.

Other line/port values do not allow the Network Server to identify the serving Application Server cluster.

©2019 CISCO SYSTEMS, INC.

Section *4.26 Network Server Redirection for REGISTER* shows a message flow that shows the REGISTER redirection by the Network Server.

©2019 CISCO SYSTEMS, INC.

3.39 Connected Line Identification Presentation (COLP)

The 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.

3.39.1 Cisco BroadWorks Sending Connected Line ID

When Connected Line Identification Presentation (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 the connected line ID depends on the setting of the *privacyVersion* SIP system parameter. Note that IP Multimedia Subsystem (IMS) mode and 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 the connected line ID is provided via the *P-Asserted-Identity* header and the *Privacy* header.
  - Because there are devices that cannot handle the *P-Asserted-Identity* header or *Privacy* header in SIP responses, an administrator can set the SIP parameter *disableCOLPForRFC3323* to "true" to disable this functionality on Cisco BroadWorks. When the functionality is disabled, Cisco BroadWorks omits the *P-Asserted-Identity* header and *Privacy* header in SIP responses. The default value for *disableCOLPForRFC3323* is "false".
- If the *privacyVersion* system parameter is set to "privacy-03" or "privacy-00", then the connected line ID is provided via the *Remote-Party-ID* header.
- If the *privacyVersion* system parameter is set to any other value, then the connected line ID is not provided in the SIP messages.

The COLP included in the *PAI/RPID* header is always a SIP URI entry. If the connected line ID is an E.164 phone number, is being sent to a user's access device without the "E164 Capable" device option, and the phone number's country code matches the user's country code, then the phone number is normalized to the appropriate prefixed national format for the country code. Otherwise, the connected line ID is sent without normalization.

The current COLP for a call is only provided in the message sent to a user's device if they have the COLP service enabled for the call and the *privacyVersion* system parameter setting allows for COLP to be included. If they do not have the COLP service enabled for the call, then the *PAI/RPID* header for COLP is not included in the responses.

3.39.2 Cisco BroadWorks Receiving Connected Line ID

The connected line ID received for a user is always ignored. The Application Server always uses the appropriate configured identity for its users and does not allow a user's device to override it.

©2019 CISCO SYSTEMS, INC.

3.40 AccessCode SIP Header

Cisco BroadWorks offers optional support of the *SIP AccessCode* header.

The 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.

3.40.1 Header Syntax

The *AccessCode* header may be included in an initial SIP INVITE request.

The *AccessCode* header has the following syntax.

```
AccessCode = "AccessCode" HCOLON gen-value
```

The following is an example of the *AccessCode* header.

```
AccessCode:1234
```

3.40.2 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. Internal calls that are routed directly to the terminating user are treated as terminations as described in section *3.40.3 Terminations*.

When this feature is disabled, the *AccessCode* header is ignored.

3.40.3 Terminations

When this feature is activated, the *AccessCode* header is included in all initial INVITEs sent to an access device for a user termination. The *AccessCode* header is set to the value configured for the *terminatingAccessCode* system parameter.

When this feature is disabled, the *AccessCode* header is not included.

3.40.4 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.
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. If the call is an internal call that is not sent to the network, then it is treated as a user termination as described in section *3.40.3 Terminations*.
When this feature is disabled, the *AccessCode* header is not included.

©2019 CISCO SYSTEMS, INC.

3.41 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 *BroadWorks AS Mode ISC Interface Specification* [64]. 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*.

3.41.1 P-Called-Party-ID Header

The Cisco BroadWorks Application Server supports the *P-Called-Party-ID* header as follows:

- 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.

The following is an example of this header.

```
P-Called-Party-ID: sip:user1-business@example.com
```

Cisco BroadWorks Application Server optionally inserts a new *P-Called-Party-ID* header in the outgoing request message. The content of the header reflects the terminating user identity that is reached. This identity can be the user's main address or one of his alternate addresses. If the Application Server has received a *P-Called-Party-ID* header in the incoming request, the latter is discarded and the newly built one is sent along with the outgoing termination request.

3.41.2 P-Access-Network-Info Header

3.41.2.1 Originating Session

Cisco BroadWorks may accept a *P-Access-Network-Info* header from an INVITE request from an access device. Depending on the configuration, Cisco BroadWorks may pass the value of this header as received to the terminating session. Cisco BroadWorks also allows a physical location to be configured for a Device Profile, and it may pass this configured value to the terminating session as the value of the *P-Access-Network-Info* header. The value of the *P-Access-Network-Info* header passed to the terminating session depends on whether the access device is a trusted device. The following table indicates the value passed to the terminating session, based on various factors.

Access Device Trusted?	PANI received in INVITE?	Provisioned physical location?	Value passed to terminating session
yes or no	no	no	none
yes or no	yes	no	received value
yes or no	no	yes	provisioned value
no	yes	yes	provisioned value
yes	yes	yes	received value

©2019 CISCO SYSTEMS, INC.

As shown in the table, if the access device is trusted, then the received *P-Access-Network-Info* value has precedence over the provisioned physical location. In contrast, if the device is untrusted, then the provisioned physical location has precedence.

Cisco BroadWorks also supports various screening services, such as Physical Location, which use the contents of the *P-Access-Network-Info* header.

3.41.2.2 Terminating Session

When Cisco BroadWorks sends an INVITE request to an access device, it adds the *P-Access-Network-Info* header only when the originating session provided a value and the access device is trusted. If the access device is untrusted, Cisco BroadWorks always omits the *P-Access-Network-Info* header.

3.41.3 Other Headers

3.41.3.1 P-Charging-Function-Addresses Header

Cisco BroadWorks recognizes the *P-Charging-Function-Addresses* header but does not proxy it in standalone mode.

3.41.3.2 P-Charging-Vector Header

Cisco BroadWorks recognizes the *P-Charging-Vector* header but does not proxy it in standalone mode.

3.41.3.3 P-Associated-URI Header

Cisco BroadWorks does not recognize the *P-Associated-URI* header.

3.41.3.4 P-Visited-Network-ID

Cisco BroadWorks does not recognize the *P-Visited-Network-ID* header.

©2019 CISCO SYSTEMS, INC.

3.42 Trunk Group Identification

For terminations, Cisco BroadWorks optionally adds *tgrp* and *trunk-context* URI parameters to the *Request-URI* for trunk group calls. For originations, it checks for them in the *Contact* header. These parameters indicate the terminating trunk group (*Request-URI*) or the originating trunk group (*Contact* header). The syntax of the parameters follows *RFC 4904* [50]. The Application Server, however, generates only SIP URIs. It never generates TEL URIs. Cisco BroadWorks expects a domain name in the *trunk-context* parameter and does not support a number in the *trunk-context* parameter.

Cisco BroadWorks also supports the *otg* and *dtg* URI parameters in SIP URIs. The Application Server uses the *dtg* parameter for Enterprise Trunk terminations to indicate the destination trunk group. The *dtg* parameter is contained in the *Request-URI*. The Application Server uses the *otg* parameter to identify the originating trunk group in trunk group originations. The *otg* parameter is contained in the URI in the *P-Asserted-Identity* header or in the *From* header. The Application Server may also use the *otg* parameter in the *Diversion* header to identity trunk groups in Out-of-Dialog PBX deflection scenarios.

When comparing the *otg* parameter value to a provisioned value, the Application Server performs a case-insensitive comparison. The *dtg* and *otg* parameters are SIP URI parameters, as defined in section 19.1.1 of *RFC 3261*.

Trunk Group identification is used in SIP INVITE requests. Cisco BroadWorks does not support trunk group identification in SIP SUBSCRIBE or SIP REGISTER requests. For SIP REGISTER, if the *tgrp* and *trunk-context* are present in the Contact, no special processing is made: they are copied transparently when the Contact target is used.

3.42.1 Message Example

If the provisioned trunk group identity is TrunkGroupA@example.net, then the parameters are formed as shown in the following examples. Note that the parameters are included in the user part of the SIP URI.

```
INVITE sip:3015551001;tgrp=TrunkGroupA;trunk-
context=example.net@192.168.40.14 SIP/2.0
Via:SIP/2.0/UDP 192.168.28.78;branch=z9hG4bK-BroadWorks.192.168.28.78-
192.168.40.14V5060-0-284840360-1474537254-1228930327375-
From:"Line A"<sip:7035553001@192.168.28.78;user=phone>;tag=1474537254-
1228930327375-
To:"James Bartel"<sip:3015551001@example.net>
Call-ID:BW123207375101208-1343479044@192.168.28.78
CSeq:284840360 INVITE
Contact:<sip:192.168.28.78:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
Accept:multipart/mixed,application/media_control+xml,application/sdp
Supported:
Max-Forwards:10
Content-Type:application/sdp
Content-Length:157

v=0
o=BroadWorks 9 1 IN IP4 192.168.40.145
s=-
c=IN IP4 192.168.40.145
t=0 0
m=audio 16384 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000/1
a=rtpmap:8 PCMA/8000/1
```

```
INVITE sip:7035553001@wgms.test;user=phone SIP/2.0
```


©2019 CISCO SYSTEMS, INC.

```
Call-ID: b3f5bb0f1679fde711ba6f568a2185b3@192.168.40.14
CSeq: 1 INVITE
From: <sip:3015551001@wgms.test;user=phone>;tag=0CB1C18048C77E31
To: <sip:7035553001@wgms.test;user=phone>
Via: SIP/2.0/UDP
192.168.40.14:5060;branch=z9hG4bKfde641967396b1bd19cb38d3d8d4acf5
Max-Forwards: 69
Contact: <sip:3015551001;trgp=TrunkGroupA;trunk-
context=example.net@192.168.40.14:5060>
Content-Type: application/sdp
Content-Length: 208

v=0
o=- 269047 269047 IN IP4 192.168.40.22
s=-
c=IN IP4 192.168.40.22
t=0 0
m=audio 16410 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

If the provisioned trunk group otg/dtg identity is TrunkGroupA, then the *otg* parameter is formed as shown in the following example.

```
INVITE sip:7035553001@example.net;user=phone SIP/2.0
Call-ID: 354fcd6c3b71df4d7eb8560eaa9fe5f2@192.168.40.14
CSeq: 1 INVITE
From:
<sip:3015551001@example.net;otg=TrunkGroupA;user=phone>;tag=2E6015F4A61B9
50A
To: <sip:7035553001@example.net;user=phone>
Via: SIP/2.0/UDP
192.168.40.14:5060;branch=z9hG4bKfc64a74d997caab55130ad3e3a4236c6
Max-Forwards: 69
Contact: <sip:3015551001@192.168.40.14:5060>
Content-Type: application/sdp
Content-Length: 206

v=0
o=- 93254 93254 IN IP4 192.168.40.22
s=-
c=IN IP4 192.168.40.22
t=0 0
m=audio 16408 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

The *dtg* is formed as shown in the following example.

```
INVITE sip:3015551001@192.168.40.14;dtg=TrunkGroupA SIP/2.0
Via:SIP/2.0/UDP 192.168.28.78;branch=z9hG4bK-BroadWorks.192.168.28.78-
192.168.40.14V5060-0-284930227-963019659-1228930507109-
From:"Line A"<sip:7035553001@192.168.28.78;user=phone>;tag=963019659-
1228930507109-
To:"James Bartel"<sip:3015551001@example.net>
Call-ID:BW123507109101208-1949285895@192.168.28.78
```


©2019 CISCO SYSTEMS, INC.

```
CSeq:284930227 INVITE
Contact:<sip:192.168.28.78:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
Accept:multipart/mixed,application/media_control+xml,application/sdp
Supported:
Max-Forwards:10
Content-Type:application/sdp
Content-Length:158

v=0
o=BroadWorks 11 1 IN IP4 192.168.40.145
s=-
c=IN IP4 192.168.40.145
t=0 0
m=audio 16384 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000/1
a=rtpmap:8 PCMA/8000/1
```


©2019 CISCO SYSTEMS, INC.

3.43 Via Header

Cisco BroadWorks constructs the *Via* header according to rules specified in *RFC 3261*. 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.

©2019 CISCO SYSTEMS, INC.

3.44 SIP Join Header (RFC 3911)

In general, Cisco BroadWorks does not support receiving a *Join* header in an INVITE request. Cisco BroadWorks rejects the INVITE request with a 481 error code. However, Cisco BroadWorks supports INVITE requests with a *Join* header in some Shared Call Appearance scenarios. For information on the use of the *Join* header for Shared Call Appearance, see the *BroadWorks SIP Access Side Extensions Interface Specification* [43].

©2019 CISCO SYSTEMS, INC.

3.45 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 *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 UAC rather than to 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

©2019 CISCO SYSTEMS, INC.

Cisco BroadWorks considers the following tags to be known tags, which cannot be transparently proxied.

100rel	early-session
199	broadworkscalltypequery
pref	gin
timer	altc
eventlist	record-aware

NOTE: Cisco BroadWorks does not validate the syntax or semantics 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 *Figure 70*. 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.

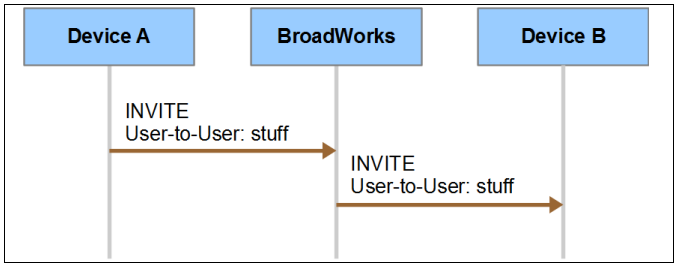

Figure 70 Transparent Proxying of an Unrecognized SIP Header

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 70*, Cisco BroadWorks could 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 71*. 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, say, the called Cisco BroadWorks user has Simultaneous Ring enabled, Device B is the user's primary access device (for example, office phone), and Device C is a network device (for example, home phone).

©2019 CISCO SYSTEMS, INC.

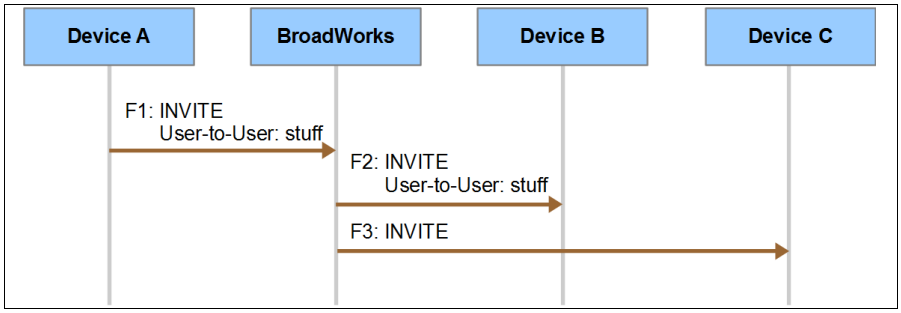

Figure 71 Transparent Proxying Depending on Destination.

Cisco BroadWorks header proxying policies may be configured 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 72*. 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.

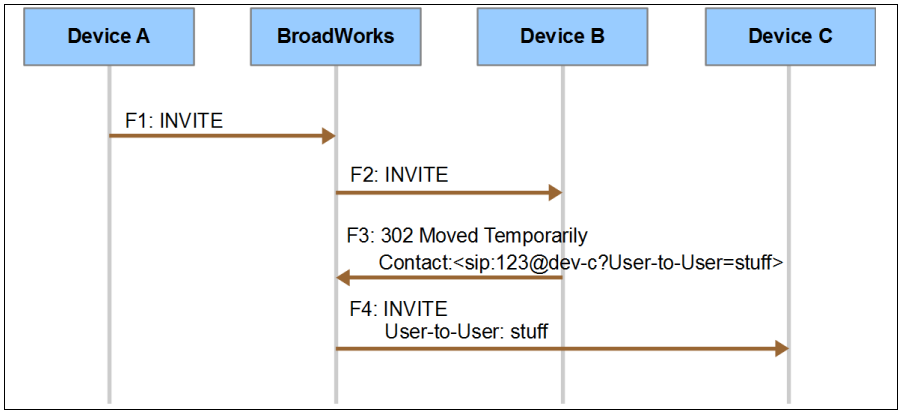

Figure 72 Header Injection from 302 Response.

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 72*, 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 73*. 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.

©2019 CISCO SYSTEMS, INC.

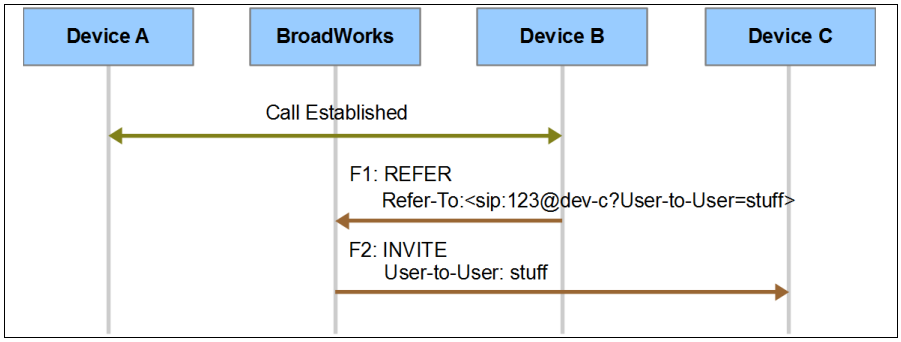

Figure 73 Header Injection from REFER Request

©2019 CISCO SYSTEMS, INC.

3.46 Advice of Charge

Cisco BroadWorks can provide Advice of Charge (AoC) information to access devices, specifically AOC-D (during a call) and AOC-E (end of call).

Cisco BroadWorks sends AoC information to access devices in message bodies encoded as *application/vnd.etsi.aoc+xml*. The Application Server may send this body type in INFO requests for AoC-D and in BYE or *200 OK* messages for AoC-D and AoC-E. It can also be used in a 487 Request Terminated response on a terminated call leg for both AoC-D and AoC-E.

The syntax for the *application/vnd.etsi.aoc+xml* is defined in *3GPP TS 24.647 v8.0.0 Advice of Charge (AoC)* [52]. Following is an example body for AoC-D.

```
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>
```

Following is an example body for AoC-E.

```
Content-Type:application/vnd.etsi.aoc+xml
Content-Length:317

<?xml version="1.0" encoding="UTF-8"?>
<aoc>
  <aoc-e>
    <recorded-charges>
      <recorded-currency-units>
        <currency-id>EUR</currency-id>
        <currency-amount>1</currency-amount>
      </recorded-currency-units>
    </recorded-charges>
    <billing-id>normal-charging</billing-id>
  </aoc-e>
</aoc>
```


©2019 CISCO SYSTEMS, INC.

3.47 Call Center Call Information

For Call Center call termination to an agent, Cisco BroadWorks optionally sends additional call information related to the Call Center call within the Call Center message body.

The Call Center call information is included in the initial INVITE request in a body of type *application/x-broadsoft-call-center*, containing an XML message. If the INVITE request contains an offer SDP, then the message bodies are included in a multipart message body per *RFC 2046*.

The following is the *Call-Center MIME* format.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://schema.broadsoft.com/as-call-center"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ascallcenter="http://schema.broadsoft.com/as-call-center"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:annotation>
    <xsd:documentation>
      BroadWorks Call Center Call Information
    </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="CallCenterCallInformation">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="waitTime" type="xsd:int">
          <xsd:annotation>
            <xsd:documentation>
              The wait time (in seconds) of the call in a call center
              before being offered to the agent.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="ccUserId" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>
              The identifier of the call center.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="callCenterName" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>
              The name of the call center.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="numCallsInQueue" type="xsd:int">
          <xsd:annotation>
            <xsd:documentation>
              The number of calls remaining in the call center (not
              counting the call being offered to the agent).
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="longestWaitingTime" type="xsd:int"
                     minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>
              The longest waiting time (in seconds) among the calls
              remaining in the call center.  Present only if there are
```


©2019 CISCO SYSTEMS, INC.

```
              calls in the call center.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

The following is an example of an initial INVITE request that contains an offer SDP and a message body of type *application/x-broadworks-call-center*.

```
INVITE sip:5146984501@192.168.8.31 SIP/2.0
Via:SIP/2.0/UDP 192.168.8.250;branch=z9hG4bKBroadWorks.-1su2ia8-
192.168.8.31V5060-0-1026496748-1075350326-1245446025687-
From:"john1
north"<sip:5146984501@192.168.8.250;user=phone>;tag=1075350326-
1245446025687-
To:"john1 north"<sip:5146984501@mtlasdev84.net>
Call-ID:BW171345687190609362467555@192.168.8.250
CSeq:1026496748 INVITE
Contact:<sip:192.168.8.250:5060>
Supported:100rel
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY,UPDATE
Accept:multipart/mixed,application/media_control+xml,application/sdp
Max-Forwards:10
Content-Type:multipart/mixed;boundary=UniqueBroadWorksBoundary
Content-Length:464
MIME-Version:1.0

--UniqueBroadWorksBoundary
Content-Type:application/sdp
Content-Length:270

v=0
o=BroadWorks 3722 1 IN IP4 192.168.8.31
s=-
c=IN IP4 192.168.8.31
t=0 0
a=sendrecv
m=audio 2240 RTP/AVP 18 0 8 101
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
--UniqueBroadWorksBoundary
Content-Type:application/x-broadworks-call-center+xml
Content-Length:146

<?xml version="1.0" encoding="ISO-8859-1"?>
<CallCenterCallInformation xmlns="http://schema.broadsoft.com/as-call-
center">
     <waitTime>426</waitTime>
     <ccUserId>callcenter@ mtlasdev84.net</ccUserId>
     <callCenterName>Call Center Premium</callCenterName>
     <numCallsInQueue>14</numCallsInQueue>
     <longestWaitingTime>415</longestWaitingTime>
</CallCenterCallInformation>
--UniqueBroadWorksBoundary--
```


©2019 CISCO SYSTEMS, INC.

3.48 Cisco BroadWorks Service Control

Cisco BroadWorks uses proprietary extensions to invoke advanced service capabilities. A request to invoke an advanced service originates from a SIP phone and is sent to the Cisco BroadWorks Application Server. The service being invoked is specified as a dedicated SIP URI.

For out-of-dialog requests, the dedicated URI is specified in the *Request-URI* of an INVITE request.

The *Request-Line* has the following syntax.

```
Request-Line   = Method SP Request-URI SP SIP-Version CRLF
```

The *Request-URI* has the following syntax when the service control URI is present.

```
Request-Line   = SIP-URI / SIPS-URI / absoluteURI
SIP-URI        = "sip:" reserved-uri "@" as-address uri-parameters
SIPS-URI       = "sips:" reserved-uri "@" as-address uri-parameters
```

For in-dialog requests, the dedicated URI is specified in the *X-BroadWorks-Service-Control* header of an INFO request.

The *X-BroadWorks-Service-Control* has the following syntax.

```
X-BroadWorks-Service-Control = "X-BroadWorks-Service-Control" HCOLON
"sip:" reserved-uri "@" as-address * (SEMI uri-params)

reserved-uri   = This is the reserved-uri defined for the service
                 being invoked.
as-address     = The address of the Application Server. This may
optionally include a port number.  This is the same address used for the
generation of INVITE requests to initiate new calls.
uri-params     = Per RFC 3261, this is the list of parameters specific to
the service being invoked.
```

3.48.1 Call Center Emergency Escalation

This control is used by a call center agent to contact a supervisor immediately. The caller is not put on hold when the emergency call is placed by the agent. Instead, the supervisor is immediately conferenced into the call.

Since it is desirable to not hold the caller, the agent typically presses a key on his phone to make an emergency escalation call, which, in turn, sends an in-dialog SIP INFO request to the Application Server to place an emergency escalation call. The service control URI contains the reserved-uri *emergency-escalation@as.domain.net*, where *as.domain.net* is the Application Server FQDN or IP address.

The uri-parameter *supervisor* is optionally added to indicate the specific supervisor to whom the call must be placed. The supervisor, if present, can be specified as an E.164 number, a complete phone number, the extension, or the location code along with the extension.

The syntax for the *supervisor* parameter is as follows.

```
supervisor = "supervisor" EQUAL ["+"] 1*DIGIT
```

The following is an example of a SIP INFO request invoking an emergency escalation call with the supervisor parameter present.

©2019 CISCO SYSTEMS, INC.

```
INFO sip:192.168.40.94:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.40.94;
  branch=z9hG4bKBroadWorks.-1t1hj8i-192.168.40.11V5060-0-8012A1305363542-
To:<sip:192.168.40.94;user=phone>;tag=1305363542-1258481993750-
From:"1 gstream"<sip:3015152010@callcenter.test>;tag=70250d479d5228b7
Call-ID:BW131953750171109114390447 4@192.168.40.94
CSeq:200 INFO
Contact: sip:sipp@192.168.40.95:5080
X-BroadWorks-Service-Control:
  sip:emergency-escalation@as.domain.net;supervisor=2131001000
Content-Length: 0
```

The following is an example of a SIP INFO request invoking an emergency escalation call with the supervisor parameter absent.

```
INFO sip:192.168.40.94:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.40.94;
  branch= z9hG4bK BroadWorks.-1t1hj8i-192.168.40.11V5060-0-28288012A1305-
To:<sip:192.168.40.94;user=phone>;tag=1305363542-1258481993750-
From:"1 gstream"<sip:3015152010@callcenter.test>;tag=70250d479d5228b7
Call-ID:BW131953750171109114390447 4@192.168.40.94
CSeq:200 INFO
Contact: sip:sipp@192.168.40.95:5080
X-BroadWorks-Service-Control:sip:emergency-escalation@as.domain.net
Content-Length: 0
```

3.48.2 Customer Originated Trace

The Customer Originated Trace functionality issues a trace to the service provider for the last incoming call for a user. The Customer Originated Trace notification contains the name and address details of the user and the caller, the timestamp of the call, the call ID, and the system ID. This information is useful if the user wishes to track an obscene, harassing, or threatening call after the call.

Cisco BroadWorks Service Control provides the ability to invoke Customer Originated Trace mid-call, typically by pressing a key on a SIP phone. The notification generated after receiving this event is for the current call.

To invoke Customer Originated Trace mid-call, the device sends a SIP INFO request to the Application Server with a service control URI in the *X-BroadWorks-Service-Control* header. The service control URI to be used to invoke mid-call Customer Originated Trace functionality is *customer-originated-trace@as.domain.net*, where *as.domain.net* is the Application Server FQDN or IP address.

Customer Originated Trace can also be invoked out-of-dialog using an INVITE with a service control URI containing the reserved URI for a Customer Originated Trace (*customer-originated-trace@as.domain.net*).

The following is an example of a SIP INFO request containing the service-control URI for a customer-originated trace.

```
INFO sip:192.168.40.94:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.40.94;
  branch=z9hG4bKBroadWorks.-1t1hj8i-192.168.40.11V5060-0-28288012A130536-
To:<sip:192.168.40.94;user=phone>;tag=1305363542-1258481993750-
From:"1 gstream"<sip:3015152010@callcenter.test>;tag=70250d479d5228b7
Call-ID:BW131953750171109114390447 4@192.168.40.94
CSeq:200 INFO
Contact: sip:sipp@192.168.40.95:5080
X-BroadWorks-Service-Control: sip:customer-originated-trace@192.168.40.94
```


©2019 CISCO SYSTEMS, INC.

```
Content-Length: 0
```

The following is an example of a SIP INVITE request invoking a customer originated trace out of dialog.

```
INVITE sip:customer-originated-trace@192.168.40.94 SIP/2.0
Via:SIP/2.0/UDP 192.168.40.13:5060
From:<sip:3015152000@callcenter.test;user=phone>;tag=643928537
To:<sip:customer-originated-trace@192.168.40.94;user=phone>
Call-ID:1769085831@192.168.40.13
CSeq: 1 INVITE
Contact:sip:3015152000@192.168.40.13:5060;user= phone;transport=udp
Expires: 300
Allow: ACK, BYE, CANCEL, INVITE, NOTIFY, OPTIONS, REFER, REGISTER
Supported: replaces
Content-Length: 257
Content-Type: application/sdp

v=0
o=3015152000 760387 IN IP4 192.168.40.13
s=ATA186 Call
c=IN IP4 192.168.40.13
t=0 0
m=audio 16384 RTP/AVP 0 4 8 101
a=rtpmap:0  PCMU/8000/1
a=rtpmap:4 G723/8000/1
a=rtpmap:8 PCMA/8000/1
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

3.48.3 Disposition Code

The Disposition Codes feature gives a Call Center agent the possibility to enter disposition codes for a Call Center call. The purpose of these codes is to associate a call with a marketing promotion or other elements.

There are two situations in which the Call Center agent has the capability to enter a Call Center call: either during an ongoing call center call, or once the Call Center call has been release and the agent in wrap-up state.

During the Call Center call, the agent phone may issue a SIP INFO request containing an *X-BroadWorks-Service-Control* header. The service control URI contains the disposition code. The SIP INFO request is issued in the context of a dialog (Call Center call) and therefore the specified disposition code is applied to that Call Center call.

The reserved URI for disposition codes is *cc-disposition-code@as.domain.net*. A new uri-parameter *code* is added to indicate the specific disposition code that is used to tag the Call Center call. This parameter is mandatory.

The syntax for the code parameter is as follows.

```
code = "code" EQUAL 1*(DIGIT / ALPHA)
```


©2019 CISCO SYSTEMS, INC.

The following is an example of a SIP INFO request requesting to tag a call with disposition code identified by “a1”.

```
INFO sip:192.168.40.94:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.40.94;
  branch= z9hG4bK BroadWorks.-1t1hj8i-192.168.40.11V5060-0-28288012A1305-
To:<sip:2131001000@192.168.40.94;user=phone>;tag=1305363542-125848199375-
From:<sip:3015152010@callcenter.test>; tag=70250d479d5228b7
Call-ID:BW131953750171109114390447 4@192.168.40.94
CSeq:200 INFO
Contact: sip:sipp@192.168.40.95:5080
X-BroadWorks-Service-Control:
  sip:cc-disposition-code@as.domain.net;code=a1
Content-Length: 0
```

Once the Call Center call has been released, the Call Center agent terminal may issue a SIP INVITE request containing a specific *Request-URI*. The SIP INVITE is issued in out-of-dialog context and the service control URI is present in the *Request-URI*. In that case the Call Center agent must be in wrap-up state and the saved call identification is used to find the identity of the Call Center call to tag. As in the case of the SIP INFO request the service control URI contains the disposition code.

The following is an example of a SIP INVITE request requesting to tag a call with disposition code identified by “b2”.

```
INVITE sip:cc-disposition-code@as.domain.net;code=b2 SIP/2.0
Via:SIP/2.0/UDP 192.168.40.13:5060;
  branch= z9hG4bK BroadWorks.-1t1hj8i-192.168.40.11V5060-0-28288012A1305-
From:<sip:3015152000@callcenter.test;user=phone>;tag=643928537
To:<sip:cc-disposition-code@as.domain.net;code=b2>
Call-ID:1769085831@192.168.40.13
CSeq: 1 INVITE
Contact:<sip:3015152000@192.168.40.13:5060;user= phone;transport=udp>
Expires: 300
Allow: ACK, BYE, CANCEL, INVITE, NOTIFY, OPTIONS, REFER, REGISTER
Supported: replaces
Content-Length: 257
Content-Type: application/sdp

v=0
o=3015152000 760387 IN IP4 192.168.40.13
s=ATA186 Call
c=IN IP4 192.168.40.13
t=0 0
m=audio 16384 RTP/AVP 0 4 8 101
a=rtpmap:0  PCMU/8000/1
a=rtpmap:4 G723/8000/1
a=rtpmap:8 PCMA/8000/1
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```


©2019 CISCO SYSTEMS, INC.

3.49 BroadSoft Proprietary Headers

The section describes the proprietary headers that Cisco BroadWorks recognizes on the access interface.

3.49.1 X-BroadWorks-Client-Session-Info

This header contains application-level information that Cisco BroadWorks relays transparently between client applications. Cisco BroadWorks may receive this header, but may send it only if the device type option *supportClientSessionInfo* is enabled.

The syntax for the *X-BroadWorks-Client-Session-Info* is defined by the following ABNF.

```
X-BroadWorks-Client-Session-Info = "X-BroadWorks-Client-Session-Info"
HCOLON token

token = 1*( alphanum / "-" / "." / "!" / "%" / "*" / "_" / "+" / "`" /
"'" / "~" )
```

The following is an example.

```
X-BroadWorks-Client-Session-Info: abc1234
```

3.49.2 X-BroadWorks-Correlation-Info

See section *3.54 Call Correlation Identifier*.

3.49.3 X-BroadWorks-Remote-Party-Info

Cisco BroadWorks sends the *X-BroadWorks-Remote-Party-Info* header in certain SIP messages to provide remote party information to an access device. The information provided includes the remote user's Cisco BroadWorks user ID, primary directory number in E.164 format (if any), and primary extension (if any).

Cisco BroadWorks sends the header in the following SIP messages:

- Initial INVITE request to identify the calling party
- 18x response to indicate the alerting party
- 200 response to INVITE to indicate the answering party
- Re-INVITE request to identify the remote party (connected identity)
- UPDATE request to identify the remote party (connected identity)

Cisco BroadWorks sends the header only if the following conditions are met:

- The access device type has the *supportRemotePartyInfo* option enabled.
- The call is a group or enterprise call.

The syntax of the *X-BroadWorks-Remote-Party-Info* is defined by the following ABNF.

```
X-BroadWorks-Remote-Party-Info = "X-BroadWorks-Remote-Party-Info" HCOLON
*(SEMI remote-party-info-param)

remote-party-info-param = user-id-param / user-dn-param / generic-param

user-id-param = "userId" EQUAL quoted-string

user-dn-param = "userDn" EQUAL quoted-string
```


©2019 CISCO SYSTEMS, INC.

The following is an example.

```
X-BroadWorks-Remote-Party-Info: userId="joe.sixpack@example.net";
userDn="tel:+19726983500;ext=500;country-code=1"
```

3.49.4 X-BroadWorks-Service Control

See section *3.48 Cisco BroadWorks Service Control*.

©2019 CISCO SYSTEMS, INC.

3.50 P-Camel Headers

Cisco BroadWorks Application Server retrieves information from the following SIP headers for accounting purposes. The headers are not proxied:

- *P-CAMEL-Loc-Info*
- *P-CAMEL-MSC-Address*
- *P-CAMEL-CellIdOrLAI*

©2019 CISCO SYSTEMS, INC.

3.51 Priority and Resource-Priority SIP Headers for Emergency Calls

In North America, emergency calling has some distinct functionality that allows an operator to identify and communicate with a 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 enable the special emergency call processing functionality for a specific call, Cisco BroadWorks requires the following conditions:

- The access device that sends the INVITE request must have the *Supports Emergency Disconnect Control* device option enabled in Cisco BroadWorks.
- The access device must add one of the following SIP headers (or both headers) to the INVITE request:
  - *Resource-Priority: emgr.0* (defined in *RFC 4412* [56])
  - *Priority: emergency* (defined in *RFC 3261* [1]).

When the emergency call processing functionality is in effect for the call, Cisco BroadWorks proxies the received *Resource-Priority* or *Priority* header when the called number has been identified as an emergency number.

The following is an example of an initial INVITE with a *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.

When the originating phone receives a re-INVITE, is should start ringing to reconnect to the emergency operator.

©2019 CISCO SYSTEMS, INC.

3.52 IPv6 Support

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* [57], *RFC 5954* [59], and *RFC 5952* [58]. Note that IPv6 the normalization to canonical form applies only to addresses generated by Cisco BroadWorks, and not to values received and proxied across. Received un-normalized IPv6 addresses proxied across Cisco BroadWorks are not normalized.

Note that IPv6 scope, link-local addresses, IPv4-mapped IPv6 addresses, and IPv4-embedded IPv6 addresses are not supported.

The operational mode determines the IP version Cisco BroadWorks uses for its SIP interface. In IPv4-only mode, Cisco BroadWorks accepts SIP message and sends SIP message 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.

In dual-stack mode, Cisco BroadWorks must decide whether to use IPv4 or IPv6 when sending a SIP request to an access device. Cisco BroadWorks makes this decision based on the IP address version of the device endpoint's contact URI. For example, if a device endpoint registers and the registered contact URI contains an IPv6 address, Cisco BroadWorks sends the request to the device using IPv6. If the contact URI has a domain name instead of an IP address, Cisco BroadWorks queries the DNS for *A* records. If the name server returns one or more IPv4 addresses for the query, then Cisco BroadWorks sends the SIP request using IPv4. Otherwise, Cisco BroadWorks queries the DNS for *AAAA* records. If the name server returns one or more IPv6 addresses, then Cisco BroadWorks sends the SIP request using IPv6. Note that Cisco BroadWorks may first query the DNS for *NAPTR* or *SRV* records, but ultimately queries the DNS for *A* or *AAAA* records.

Cisco BroadWorks supports the alternate connectivity mechanism for media negotiation, as described in *RFC 6947*. With 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.

©2019 CISCO SYSTEMS, INC.

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.

3.52.1 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=z9hG4bKc42bcaae0d5ba68cd8e1704
  25087270b6ec73337,SIP/2.0/UDP
  [fd5d:e1c5:f9d8:0:2024:e8ff:fe48:7d29]:5070;branch=z9hG4bK6201;received=fd5d: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
```


©2019 CISCO SYSTEMS, INC.

```
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
```


©2019 CISCO SYSTEMS, INC.

3.53 Support for Multiple Phone Numbers in SIP (RFC 6140)

Cisco BroadWorks supports key aspects of *RFC 6140*.

Cisco BroadWorks supports “GIN” registration in connection with SIP trunking. The access device registers with the GIN mechanism with the trunk group pilot user’s line/port as the Address of Record and the address of the trunk group as the contact URI. The registration effectively enters a contact URI into Cisco BroadWorks location database for each business trunking user who is reachable via that trunk group.

Cisco BroadWorks understands the *bnc* parameter in the contact URI of the GIN registration request. However, Cisco BroadWorks does not require the *bnc* parameter, and the registration may achieve the same effect without the *bnc* parameter.

©2019 CISCO SYSTEMS, INC.

3.54 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
- NOTIFY request for these event packages: *call-info*, *call-park*, *dialog*, *talk*, *hold*, *refer*

Based on the configuration of Cisco BroadWorks, the correlation identifier can be unique within a Cisco BroadWorks cluster or globally unique.

©2019 CISCO SYSTEMS, INC.

3.55 Stateless Proxy for Geographical Redundancy

3.55.1 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 *BroadWorks Redundancy Guide* [63].

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.

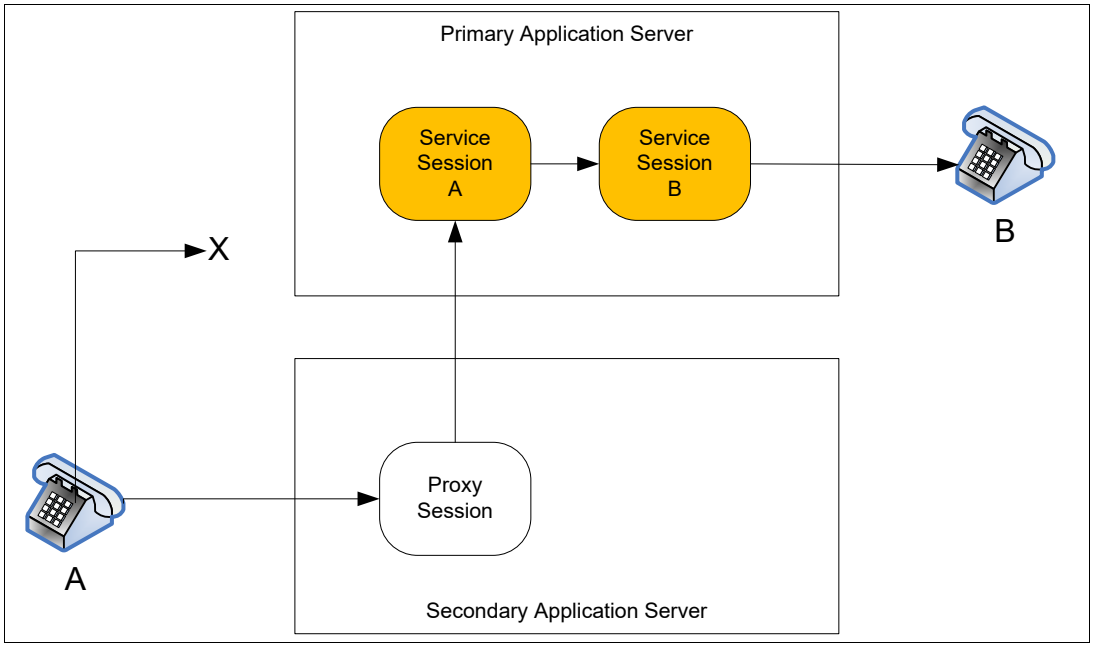

Figure 74 Stateless Proxy Server

In the 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.

©2019 CISCO SYSTEMS, INC.

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.

3.55.2 Call Flows

There are three different call scenarios in which the secondary Application Server performs as a stateless proxy server.

3.55.2.1 Scenario 1: Access Device to Secondary Application Server to Primary Application Server

The first scenario is shown in the call flow diagram in *Figure 75*. 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.

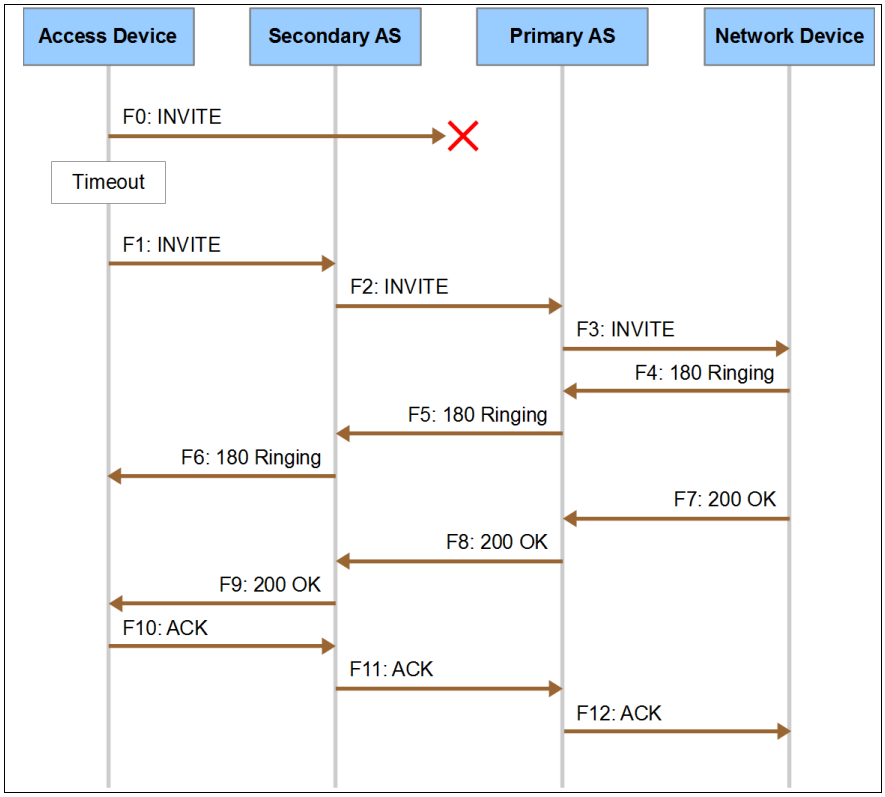

Figure 75 Proxy Scenario: Access Device Cannot Reach Primary Application Server

©2019 CISCO SYSTEMS, INC.

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 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 only appears in the leg between the primary and secondary Application Servers.

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>
```

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:10.0.40.40
```

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.

©2019 CISCO SYSTEMS, INC.

3.55.2.2 Scenario 2: Network Device to Secondary Application Server to Primary Application Server

The second scenario, shown in the call flow diagram in *Figure 76*, 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.

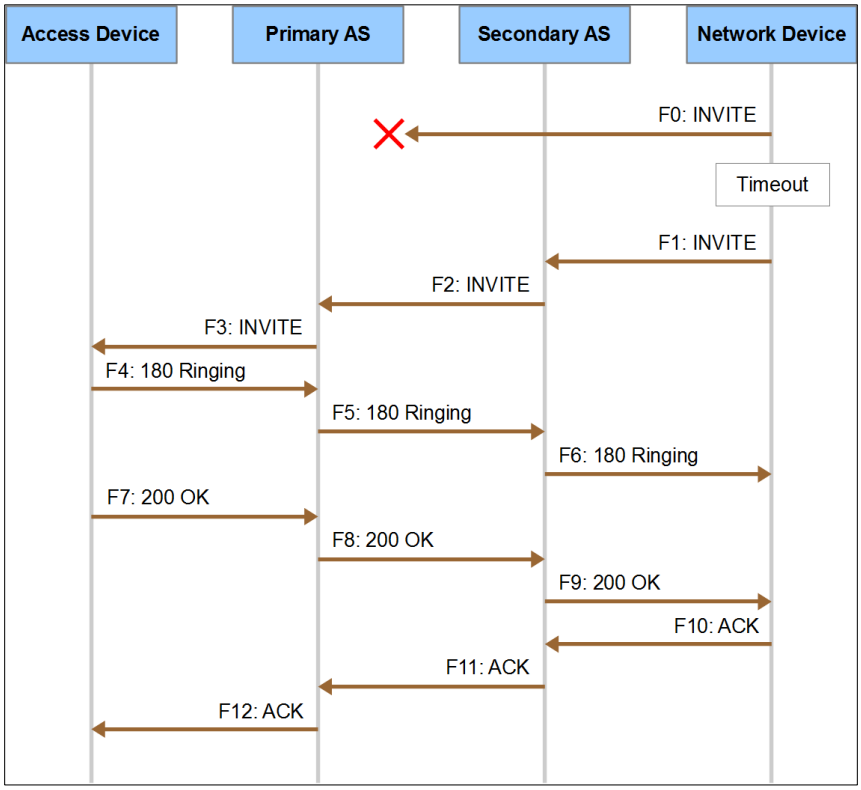

Figure 76 Proxy Scenario: Network Device Cannot Reach Primary Application Server

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 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.

©2019 CISCO SYSTEMS, INC.

- *bwpeer* – This parameter also tags the entry as one that identifies the secondary Application Server acting in the proxy server role. However, it only appears in 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.

©2019 CISCO SYSTEMS, INC.

3.55.2.3 Scenario 3: Primary Application Server to Secondary Application Server to Access Device

The third scenario is shown in *Figure 77* . 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.

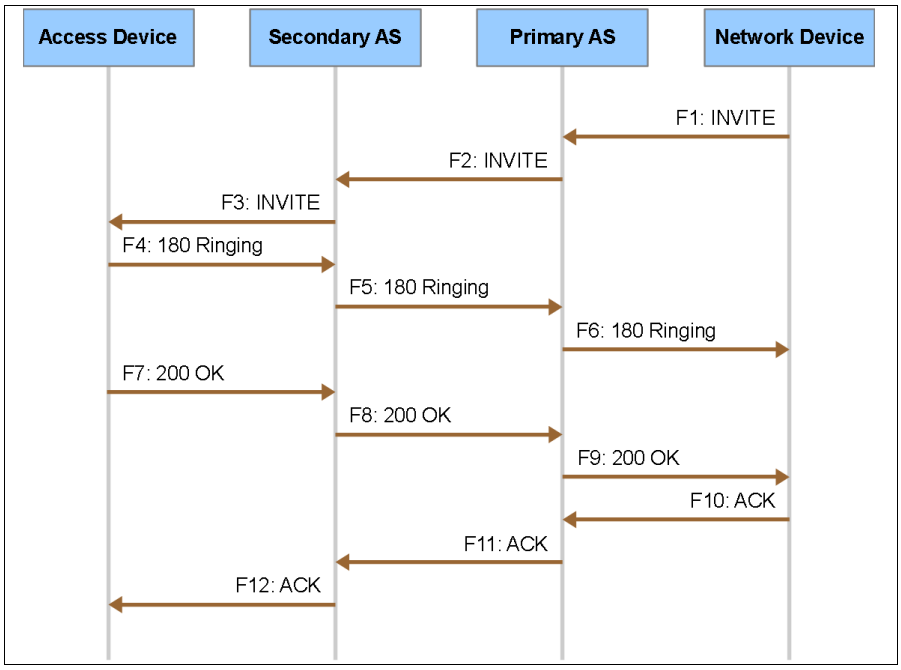

Figure 77 Proxy Scenario: Network Device Cannot Reach Primary Application Server

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 only appears in the leg between the primary and secondary Application Servers.

©2019 CISCO SYSTEMS, INC.

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.

©2019 CISCO SYSTEMS, INC.

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. 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 78*. 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.

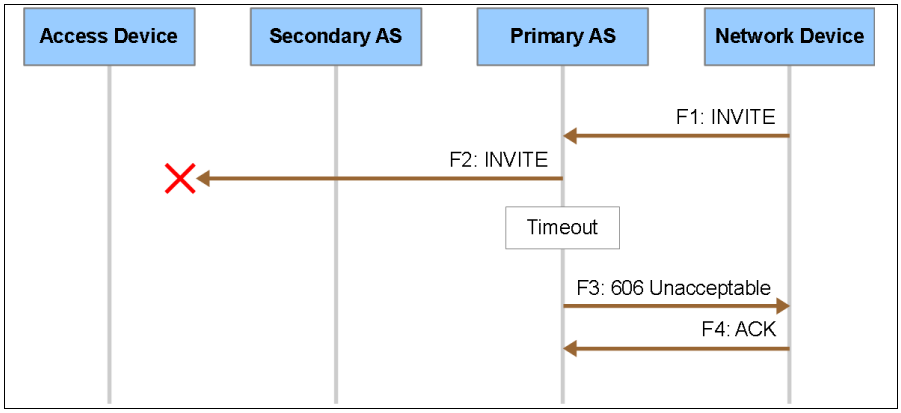

Figure 78 Call Failure Due to Unreachable Access Device

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 details, see the *BroadWorks AS Mode ISC Interface Specification* [64].

3.55.3 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 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 *BroadWorks Redundancy Guide*.

©2019 CISCO SYSTEMS, INC.

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 may 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".

3.55.4 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. The decision logic is explained in the *BroadWorks Redundancy Guide*.

3.55.4.1 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:

©2019 CISCO SYSTEMS, INC.

- the value of the *bw.sip.accessinterfaceviahost* startup parameter, if 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 set
  - otherwise, the configured private IP address, if 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 that it is responsible for (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 following table.
- 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*.

©2019 CISCO SYSTEMS, INC.

– 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. The default value is "nil". 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".

©2019 CISCO SYSTEMS, INC.

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 host name 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.

©2019 CISCO SYSTEMS, INC.

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.

3.55.4.2 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, 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. If the proxy server receives a 1xx response to an INVITE request, 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 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.

3.55.5 Peer Monitoring

Before the secondary Application Server routes an initial INVITE request to the primary Application Server (Scenario 1 or Scenario 2), 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), 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 to monitor connectivity status.

©2019 CISCO SYSTEMS, INC.

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).

3.55.6 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
```


©2019 CISCO SYSTEMS, INC.

3.55.7 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.

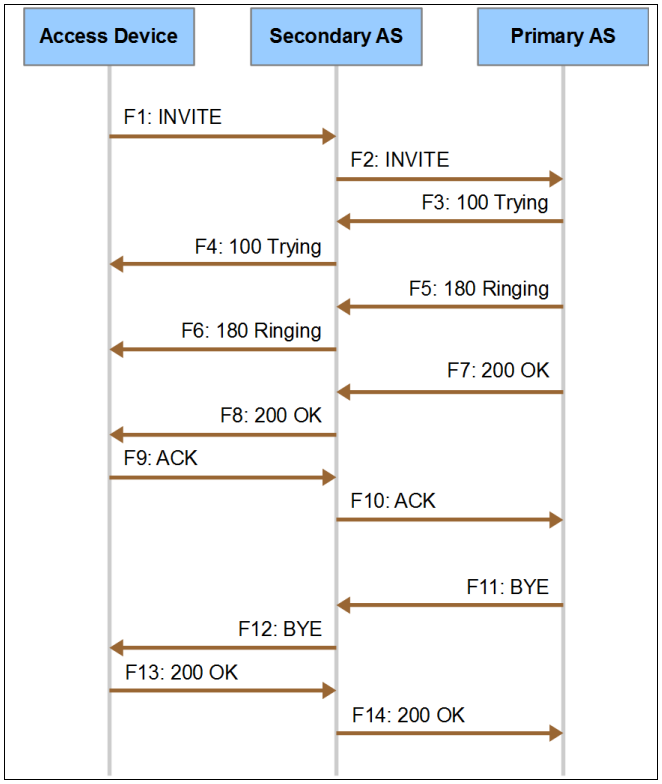

Figure 79 Call Flow Diagram for Secondary Application Server Acting as Proxy Server

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
```


©2019 CISCO SYSTEMS, INC.

```
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
```


©2019 CISCO SYSTEMS, INC.

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>
```


©2019 CISCO SYSTEMS, INC.

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
```


©2019 CISCO SYSTEMS, INC.

```
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>
```

3.55.8 Session Recording Protocol (draft-ietf-siprec-protocol-09)

Cisco BroadWorks supports many features of the Session Recording Protocol described in the *draft-ietf-siprec-protocol-09*. Specifically, on the access interface, Cisco BroadWorks recognizes the *record-aware* option tag in SIP messages, as well as the *recordpref* and *record* attributes in SDP. For more information, see the *BroadWorks Call Recording Interface Guide* [65].

©2019 CISCO SYSTEMS, INC.

3.56 Preconditions Framework (RFC 3312)

Reference Documents:

- RFC 3312: Integration of Resource Management and Session Initiation Protocol (SIP), October 2002

3.56.1 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 *3.9 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.

©2019 CISCO SYSTEMS, INC.

3.56.2 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.

©2019 CISCO SYSTEMS, INC.

3.57 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.

3.57.1 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[10], Cisco BroadWorks stores this information and may send the "video" tag in outgoing INVITE requests on behalf of the remote 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 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 generally supports and follows the requirements of *RFC 3840* [62] and IR.94 [68]. 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".

[10] The registered name of 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.

©2019 CISCO SYSTEMS, INC.

4 Call Flows

This section contains call flows of various scenarios that must be supported by an access 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 an access device to interwork with Cisco BroadWorks. The following scenarios are depicted:

- Access Device to Cisco BroadWorks Call
- Cisco BroadWorks to Access Device Call
- Access Device Releases Call
- Cisco BroadWorks Releases Call
- Access Device Registration with Authentication Challenge
- Access Device Holds Call
- Cisco BroadWorks Holds Call
- Cisco BroadWorks Initiated Media Request with SDP (used in blind transfer, transfer, conferencing, and so on)
- Cisco BroadWorks Initiated Media Request without SDP (used in blind transfer, transfer, conferencing, and so on)
- Access Device Initiated Media Request with SDP (used in blind transfer, transfer, conferencing, and so on)
- Access Device to Cisco BroadWorks Requesting Calling Party Identity Blocking
- Access Device to Cisco BroadWorks Call Requesting Calling Party Identity Blocking (*RFC 3323/3325*)
- Access Device to Cisco BroadWorks Call Requesting Calling Party Identity Blocking (draft-ietf-sip-privacy-03)
- Cisco BroadWorks to Access Device with Calling Party Identity Blocking
- Cisco BroadWorks to Access Device with Priority Alerting Information
- Access Device Initiates Blind Transfer
- Access Device Initiates Transfer with Consultation
- Cisco BroadWorks Sends Message Waiting Indication to Access Device
- Cisco BroadWorks to Access Device Call with Redirection (Diversion), Unconditional Call Forwarding
- Cisco BroadWorks Informs Access Device to Play Call-Waiting Tone
- Cisco BroadWorks Informs Access Device to Stop Call-Waiting Tone
- Access Device Informs Cisco BroadWorks that the user presses the flash hook
- Access Device to Cisco BroadWorks Subscription (generic-event event package example)
- Cisco BroadWorks to Access Device Subscription (generic-event event package example)
- Access Device to Cisco BroadWorks SMS

©2019 CISCO SYSTEMS, INC.

- Session Border Controller (on behalf of access device) to Cisco BroadWorks Registration

Each flow shows an annotated example of an access device to Cisco BroadWorks call or Cisco BroadWorks to access device call, where the Cisco BroadWorks Application Server is acting on behalf of Cisco BroadWorks.

4.1 Access Device to Cisco BroadWorks Call

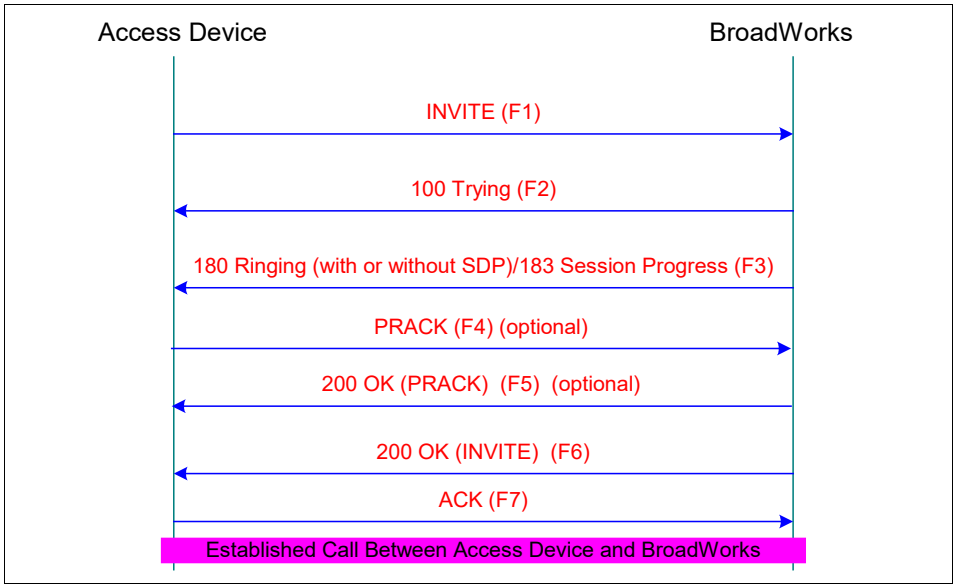

Figure 80 Access Device to Cisco BroadWorks Call

4.1.1 F1 – INVITE: Access Device to Cisco BroadWorks

```
INVITE sip:2403645138@192.168.5.253;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From: "2403649314"
<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-372ea22b
To: <sip:2403645138@192.168.5.253;user=phone>
Call-ID: 0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
Date: Tue, 04 Jun 2002 19:52:42 GMT
CSeq: 101 INVITE
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Expires: 180
Supported:100rel,timer
Content-Type: application/sdp
Max-Forwards:10
Content-Length: 170
v=0
o=SDP 26088 15595 IN IP4 192.168.5.214
s=SIP Call
c=IN IP4 192.168.5.214
t=0 0
m=audio 23890 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```


©2019 CISCO SYSTEMS, INC.

4.1.2 F2 – 100 Trying: Cisco BroadWorks to Access Device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b352
66f-372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq:101 INVITE
Content-Length:0
```

4.1.3 F3 – 180 Ringing (with or without SDP)/183 Session Progress: Cisco BroadWorks to Access Device

```
SIP/2.0 180 Ringing
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b352
66f-372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq:101 INVITE
Require:100rel^M
RSeq:1016510814
Content-Length:0
```

4.1.4 F4 – PRACK: Access Device to Cisco BroadWorks

```
PRACK sip: 192.168.5.253;user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b352
66f-372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq: 102 PRACK
RAck:1016510814 101 INVITE
Max-Forwards:10
Content-Length: 0
```

4.1.5 F5 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b352
66f-372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq: 102 PRACK
Content-Length: 0
```

4.1.6 F6 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b352
66f-372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
```


©2019 CISCO SYSTEMS, INC.

```
CSeq:101 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Contact:<sip:192.168.5.253:5060>
Content-Type:application/sdp
Content-Length:158

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
```

4.1.7 F7 – ACK: Access Device to Cisco BroadWorks

```
ACK sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To: <sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID: 0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
Date: Tue, 04 Jun 2002 19:52:48 GMT
CSeq: 101 ACK
User-Agent: AccessDevice
Max-Forwards:10
Content-Length: 0
```

4.2 Cisco BroadWorks to Access Device Call

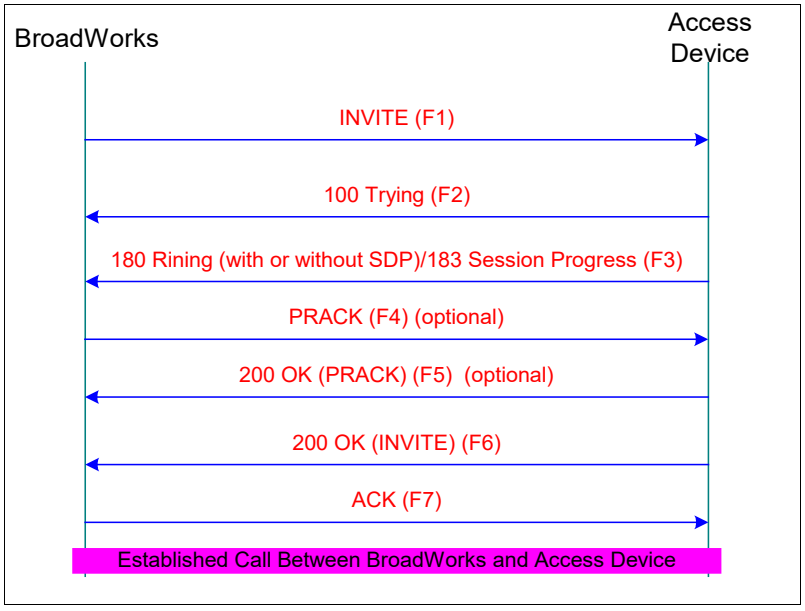

Figure 81 Cisco BroadWorks to Access Device Call

©2019 CISCO SYSTEMS, INC.

4.2.1 F1 – INVITE: Cisco BroadWorks to Access Device

```
INVITE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003454-1569407939-1023204223358
From:<sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To:"richard ricardo"<sip:2403649314@192.168.5.253;user=phone>
Call-ID:BW112343035704060202316497252421 72@192.168.5.253
CSeq:501003454 INVITE
Contact:<sip:192.168.5.253:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:158

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 18054 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.2.2 F2 – 100 Trying: Access Device to Cisco BroadWorks

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003454-1569407939-1023204223358
From: <sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To: "richard ricardo"<sip:2403649314@192.168.5.253;user=phone>
Call-ID: BW112343035704060202316497252421 72@192.168.5.253
Date: Tue, 04 Jun 2002 20:24:16 GMT
CSeq: 501003454 INVITE
Contact: sip:2403649314@192.168.5.214:5060
Content-Length: 0
```

4.2.3 F3 – 180 Ringing (with or without SDP)/183 Session Progress: Access Device to Cisco BroadWorks

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003454-1569407939-1023204223358
From: <sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To: "richard
ricardo"<sip:2403649314@192.168.5.253;user=phone>;tag=0003e3630c94001d20f
605e8-1dabcb15
Call-ID: BW112343035704060202316497252421 72@192.168.5.253
Date: Tue, 04 Jun 2002 20:24:16 GMT
CSeq: 501003454 INVITE
Require: 100rel
RSeq: 1234
Contact: sip:2403649314@192.168.5.214:5060
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

4.2.4 F4 – PRACK: Cisco BroadWorks to Access Device

```
PRACK sip:2403649314@192.168.5.214:5060 SIP/2.0
 Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003455-1569407939-1023204223358
From: <sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To: "richard
ricardo"<sip:2403649314@192.168.5.253;user=phone>;tag=0003e3630c94001d20f
605e8-1dabcb15
Call-ID: BW112343035704060202316497252421 72@192.168.5.253
CSeq: 501003455 INVITE
RAck:1234 501003454 INVITE
Max-Forwards:10
Content-Length: 0
```

4.2.5 F5 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003455-1569407939-1023204223358
From: <sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To: "richard
ricardo"<sip:2403649314@192.168.5.253;user=phone>;tag=0003e3630c94001d20f
605e8-1dabcb15
Call-ID: BW112343035704060202316497252421 72@192.168.5.253
CSeq: 501003455 INVITE
Content-Length:0
```

4.2.6 F6 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003454-1569407939-1023204223358
From: <sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To: "richard
ricardo"<sip:2403649314@192.168.5.253;user=phone>;tag=0003e3630c94001d20f
605e8-1dabcb15
Call-ID: BW112343035704060202316497252421 72@192.168.5.253
Date: Tue, 04 Jun 2002 20:24:18 GMT
CSeq: 501003454 INVITE
Contact: sip:2403649314@192.168.5.214:5060
Content-Type: application/sdp
Content-Length: 165

v=0
o=SDP 12910 22219 IN IP4 192.168.5.214
s=SDP
c=IN IP4 192.168.5.214
t=0 0
m=audio 23894 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```


©2019 CISCO SYSTEMS, INC.

4.2.7 F7 – ACK: Cisco BroadWorks to Access Device

```
ACK sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003454A1569407939-1023204223358
From:<sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To:"richard
ricardo"<sip:2403649314@192.168.5.253;user=phone>;tag=0003e3630c94001d20f
605e8-1dabcb15
Call-ID:BW112343035704060202316497252421 72@192.168.5.253
CSeq:501003454 ACK
Contact:<sip:192.168.5.253:5060>
Max-Forwards:10
Content-Length:0
```

4.3 Access Device Releases Call

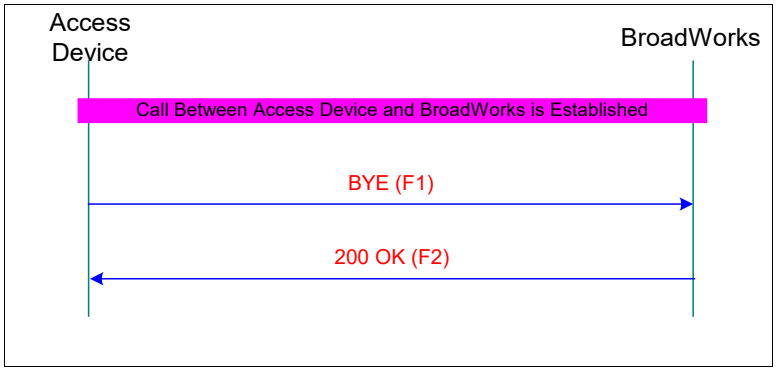

Figure 82 Access Device Releases Call

4.3.1 F1 – BYE: Access Device to Cisco BroadWorks

```
BYE sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001b3f7af847-
285c8e32
To: <sip:2403645214@192.168.5.253;user=phone>;tag=365083980-1023203926937
Call-ID: 0003e363-0c9406db-354afb46-67f9b7e3@192.168.5.214
Date: Tue, 04 Jun 2002 20:23:38 GMT
CSeq: 102 BYE
User-Agent: AccessDevice
Max-Forwards:10
Content-Length: 0
```

4.3.2 F2 – 200 OK: BroadWorks to Access Device

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001b3f7af847-
285c8e32
To: <sip:2403645214@192.168.5.253;user=phone>;tag=365083980-1023203926937
Call-ID: 0003e363-0c9406db-354afb46-67f9b7e3@192.168.5.214
CSeq: 102 BYE
```


©2019 CISCO SYSTEMS, INC.

```
Content-Length:0
```

4.4 BroadWorks Releases Call

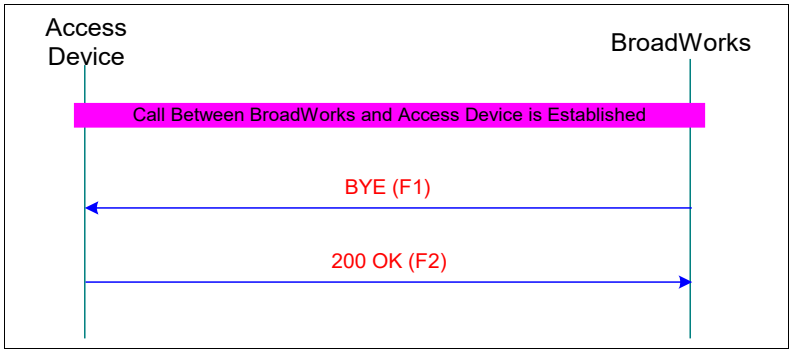

Figure 83 Cisco BroadWorks Releases Call

4.4.1 F1 – BYE: Cisco BroadWorks to Access Device

```
BYE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003455-1569407939-1023204223358
From:<sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To:"richard
ricardo"<sip:2403649314@192.168.5.253;user=phone>;tag=0003e3630c94001d20f
605e8-1dabcb15
Call-ID:BW112343035704060202316497252421 72@192.168.5.253
CSeq:501003455 BYE
Max-Forwards:10
Content-Length:0
```

4.4.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-501003455-1569407939-1023204223358
From: <sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-
1023204223358
To: "richard
ricardo"<sip:2403649314@192.168.5.253;user=phone>;tag=0003e3630c94001d20f
605e8-1dabcb15
Call-ID: BW112343035704060202316497252421 72@192.168.5.253
Date: Tue, 04 Jun 2002 20:24:30 GMT
CSeq: 501003455 BYE
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

4.5 Access Device Registration with Authentication Challenge

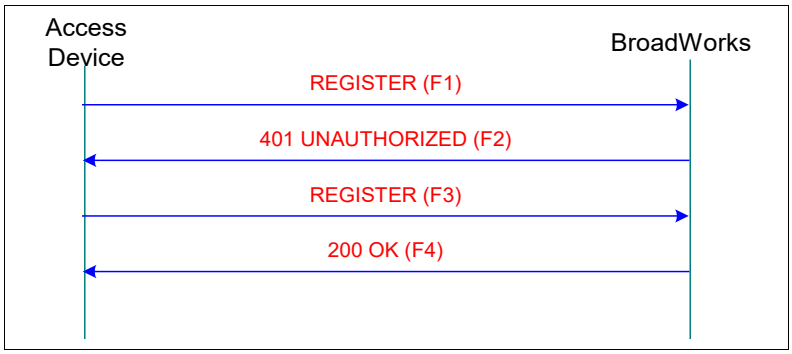

Figure 84 Access Device Registration with Authentication Challenge

4.5.1 F1 – REGISTER: Access Device to Cisco BroadWorks

```
REGISTER sip:192.168.5.253 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.87:5060
From: sip:2403649320@192.168.5.253
To: sip:2403649320@192.168.5.253
Call-ID: 0003e363-0bdf23e7-0f0bd70e-5c9e4003@192.168.5.87
Date: Mon, 03 Jun 2002 18:02:51 GMT
CSeq: 101 REGISTER
User-Agent: AccessDevice
Contact: sip:2403649320@192.168.5.87:5060
Max-Forwards:10
Content-Length: 0
Expires: 86400
```

4.5.2 F2 – 401 UNAUTHORIZED: Cisco BroadWorks to Access Device

```
SIP/2.0 401 Unauthorized
Via:SIP/2.0/UDP 192.168.5.87:5060
From:<sip:2403649320@192.168.5.253>
To:<sip:2403649320@192.168.5.253>
Call-ID:0003e363-0bdf23e7-0f0bd70e-5c9e4003@192.168.5.87
CSeq:101 REGISTER
Content-Length:0
WWW-Authenticate:DIGEST
realm="BroadWorks",algorithm=MD5,nonce="1023109332501"
```

4.5.3 F3 – REGISTER: Access Device to Cisco BroadWorks

```
REGISTER sip:192.168.5.253 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.87:5060
From: sip:2403649320@192.168.5.253
To: sip:2403649320@192.168.5.253
Call-ID: 0003e363-0bdf23e7-0f0bd70e-5c9e4003@192.168.5.87
Date: Mon, 03 Jun 2002 18:02:51 GMT
CSeq: 102 REGISTER
User-Agent: AccessDevice
Contact: sip:2403649320@192.168.5.87:5060
Authorization: Digest
username="2403649320",realm="BroadWorks",uri="sip:192.168.5.253",response
="4e4c8e53088de62e109a82dee4ad2a01",nonce="1023109332501",algorithm=MD5
Max-Forwards:10
```


©2019 CISCO SYSTEMS, INC.

```
Content-Length: 0
Expires: 86400
```

4.5.4 F4 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.87:5060
From:<sip:2403649320@192.168.5.253>
To:<sip:2403649320@192.168.5.253>
Call-ID:0003e363-0bdf23e7-0f0bd70e-5c9e4003@192.168.5.87
CSeq:102 REGISTER
Content-Length:0
Contact:<sip:2403649320@192.168.5.87:5060>;q=0.5;expires=86399
```

4.6 Access Device Holds Call

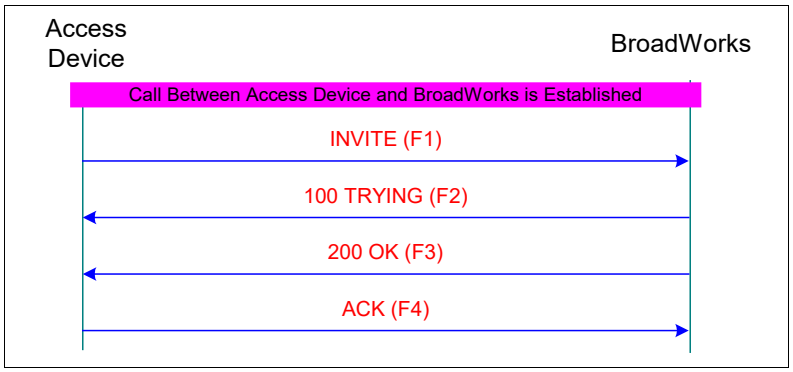

Figure 85 Access Device Holds Call

4.6.1 Access Device Holds Using RFC 2543 Hold Mechanism (c=0.0.0.0)

4.6.1.1 F1 – INVITE: Access Device to Cisco BroadWorks

```
INVITE sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
To: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:18 GMT
CSeq: 102 INVITE
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Content-Type: application/sdp
Max-Forwards:10
Content-Length: 163

v=0
o=SDP 14918 9029 IN IP4 192.168.5.214
s=SDP
c=IN IP4 0.0.0.0
t=0 0
m=audio 23900 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```


©2019 CISCO SYSTEMS, INC.

4.6.1.2 F2 – 100 Trying: Cisco BroadWorks to Access Device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3
027-31d1d958
To:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:102 INVITE
Content-Length:0
```

4.6.1.3 F3 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3
027-31d1d958
To:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:102 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Contact:<sip:192.168.5.253:5060>
Content-Type:application/sdp
Content-Length:107

v=0
o=BroadWorks 3 1 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
```

4.6.1.4 F4 – ACK: Access Device to Cisco BroadWorks

```
ACK sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
To: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:19 GMT
CSeq: 102 ACK
User-Agent: AccessDevice
Max-Forwards:10
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

4.6.2 Access Device Holds Using RFC 3264 Hold Mechanism (a=sendonly/a=inactive)

4.6.2.1 F1 – INVITE: Access Device to Cisco BroadWorks

```
INVITE sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
To: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:18 GMT
CSeq: 102 INVITE
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Content-Type: application/sdp
Max-Forwards:10
Content-Length: 163

v=0
o=SDP 14918 9029 IN IP4 192.168.5.214
s=SDP
c=IN IP4 192.168.5.214
t=0 0
m=audio 23900 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=sendonly
```

4.6.2.2 F2 – 100 Trying: Cisco BroadWorks to Access Device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3
027-31d1d958
To:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:102 INVITE
Content-Length:0
```

4.6.2.3 F3 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3
027-31d1d958
To:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:102 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Contact:<sip:192.168.5.253:5060>
Content-Type:application/sdp
Content-Length:107

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 0.0.0.0
t=0 0
```


©2019 CISCO SYSTEMS, INC.

```
m=audio 16864 RTP/AVP 0
a=inactive
```

4.6.2.4 F4 – ACK: Access Device to Cisco BroadWorks

```
ACK sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
To: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:19 GMT
CSeq: 102 ACK
User-Agent: AccessDevice
Max-Forwards:10
Content-Length: 0
```

4.7 Cisco BroadWorks Holds Call

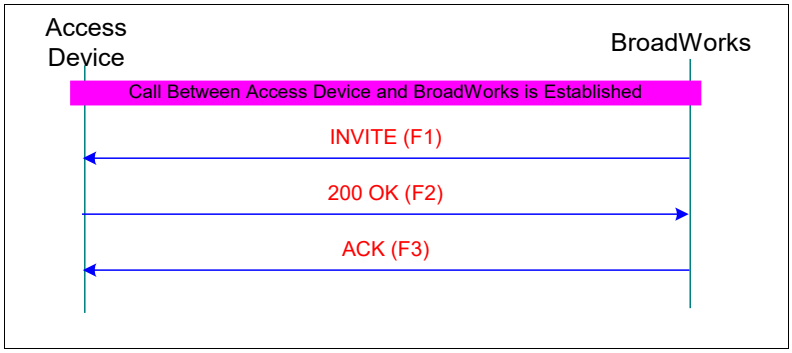

Figure 86 Cisco BroadWorks Holds Call

4.7.1 F1 – INVITE: Cisco BroadWorks to Access Device

```
INVITE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-502889295-445624321-1023207995257
From:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-
1023207995257
To:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f302
7-31d1d958
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:502889295 INVITE
Contact:<sip:192.168.5.253:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:103

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 0.0.0.0
t=0 0
```


©2019 CISCO SYSTEMS, INC.

```
m=audio 16864 RTP/AVP 0
a=inactive
```

4.7.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-502889295-445624321-1023207995257
From: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-
1023207995257
To:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:26 GMT
CSeq: 502889295 INVITE
Contact: sip:2403649314@192.168.5.214:5060
Content-Type: application/sdp
Content-Length: 162

v=0
o=SDP 6371 8649 IN IP4 192.168.5.214
s=SDP
c=IN IP4 192.168.5.214
t=0 0
m=audio 23900 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=inactive
```

4.7.3 F3 – ACK: Cisco BroadWorks to Access Device

```
ACK sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-502889295A445624321-1023207995257
From:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-
1023207995257
To:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f302
7-31d1d958
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:502889295 ACK
Contact:<sip:192.168.5.253:5060>
Max-Forwards:10
Content-Length:0
```


©2019 CISCO SYSTEMS, INC.

4.8 Cisco BroadWorks Initiated Media Request with SDP (Used in Blind Transfer, Transfer, Conferencing)

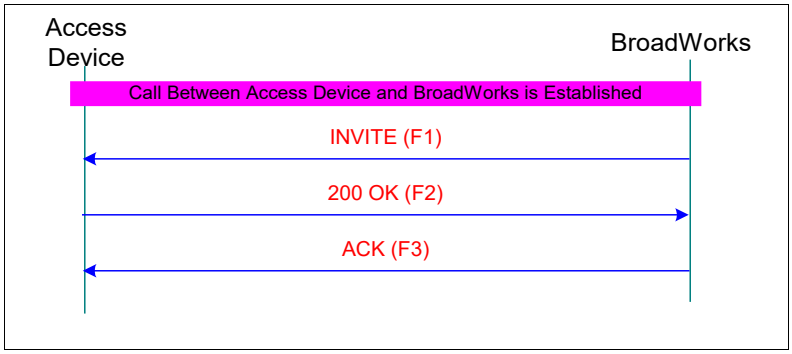

Figure 87 Cisco BroadWorks Initiated Media Request with SDP

4.8.1 F1 – INVITE: Cisco BroadWorks to Access Device

```
INVITE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-502889296-445624321-1023207995257
From:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-
1023207995257
To:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f302
7-31d1d958
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:502889296 INVITE
Contact:<sip:192.168.5.253:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:107

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 192.168.5.215
t=0 0
m=audio 16864 RTP/AVP 0
```

4.8.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-502889296-445624321-1023207995257
From: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-
1023207995257
To:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:28 GMT
CSeq: 502889296 INVITE
Contact: sip:2403649314@192.168.5.214:5060
Content-Type: application/sdp
```


©2019 CISCO SYSTEMS, INC.

```
Content-Length: 165

v=0
o=SDP 12725 12419 IN IP4 192.168.5.214
s=SDP
c=IN IP4 192.168.5.214
t=0 0
m=audio 23900 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.8.3 F3 – ACK: Cisco BroadWorks to Access Device

```
ACK sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-502889296A445624321-1023207995257
From:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-
1023207995257
To:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f302
7-31d1d958
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:502889296 ACK
Contact:<sip:192.168.5.253:5060>
Max-Forwards:10
Content-Length:0
```

4.9 Cisco BroadWorks Initiated Media Request without SDP (Used in Blind Transfer, Transfer, Conferencing)

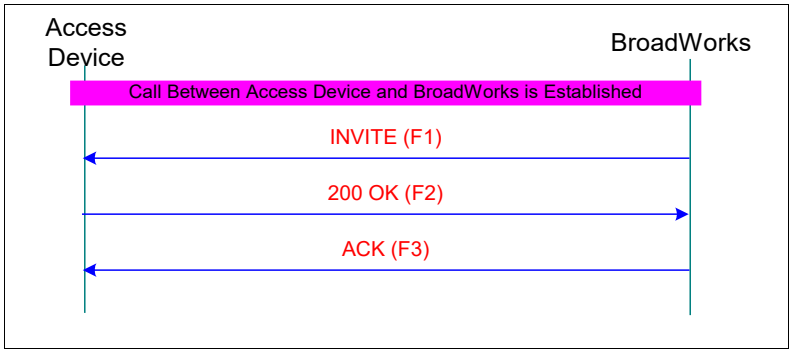

Figure 88 Cisco BroadWorks Initiated Media Request without SDP

4.9.1 F1 – INVITE: Cisco BroadWorks to Access Device

```
INVITE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-526825663-377809407-1023255867893
From:<sip:2403649311@192.168.5.253;user=phone>;tag=377809407-
1023255867893
To:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400212a32532
2-1f653ca3
Call-ID:0003e363-0c940772-564acedc-13f49430@192.168.5.214
CSeq:526825663 INVITE
Contact:<sip:192.168.5.253:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
```


©2019 CISCO SYSTEMS, INC.

```
Max-Forwards:10
Content-Length:0
```

4.9.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-526825663-377809407-1023255867893
From: <sip:2403649311@192.168.5.253;user=phone>;tag=377809407-
1023255867893
To:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400212a325322-
1f653ca3
Call-ID: 0003e363-0c940772-564acedc-13f49430@192.168.5.214
Date: Wed, 05 Jun 2002 10:45:53 GMT
CSeq: 526825663 INVITE
Contact: sip:2403649314@192.168.5.214:5060
Content-Type: application/sdp
Content-Length: 164

v=0
o=SDP 7775 11212 IN IP4 192.168.5.214
s=SDP Call
c=IN IP4 192.168.5.214
t=0 0
m=audio 23902 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```

4.9.3 F3 – ACK: Cisco BroadWorks to Access Device

```
ACK sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-526825663A377809407-1023255867893
From:<sip:2403649311@192.168.5.253;user=phone>;tag=377809407-
1023255867893
To:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400212a32532
2-1f653ca3
Call-ID:0003e363-0c940772-564acedc-13f49430@192.168.5.214
CSeq:526825663 ACK
Contact:<sip:192.168.5.253:5060>
Max-Forwards:10
Content-Length:158
Content-Type:application/sdp

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 192.168.5.215
t=0 0
m=audio 18774 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```


©2019 CISCO SYSTEMS, INC.

4.10 Access Device Initiated Media Request with SDP (Used in Blind Transfer, Transfer, Conferencing)

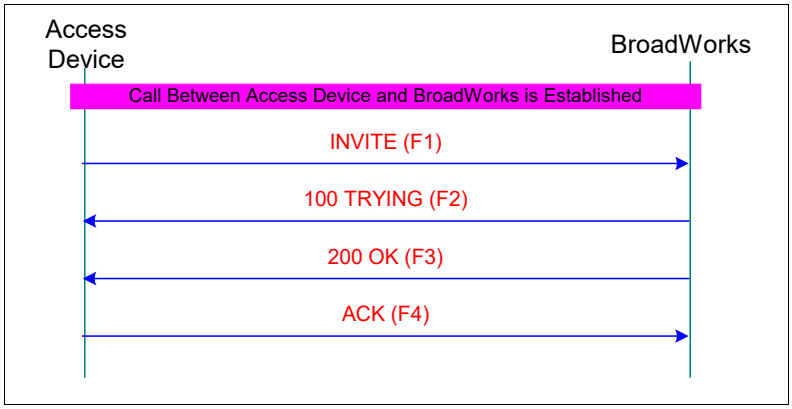

Figure 89 Access Device Initiated Media Request with SDP

4.10.1 F1 – INVITE: Access Device to Cisco BroadWorks

```
INVITE sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
To: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:22 GMT
CSeq: 103 INVITE
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Content-Type: application/sdp
Max-Forwards:10
Content-Length: 170

v=0
o=SDP 14516 11215 IN IP4 192.168.5.214
s=SDP
c=IN IP4 192.168.5.214
t=0 0
m=audio 23900 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```

4.10.2 F2 – 100 Trying: Cisco BroadWorks to Access Device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3
027-31d1d958
To:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:103 INVITE
Content-Length:0
```


©2019 CISCO SYSTEMS, INC.

4.10.3 F3 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3
027-31d1d958
To:<sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID:0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
CSeq:103 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Contact:<sip:192.168.5.253:5060>
Content-Type:application/sdp
Content-Length:107

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 192.168.5.215
t=0 0
m=audio 16864 RTP/AVP 0
```

4.10.4 F4 – ACK: Access Device to Cisco BroadWorks

```
ACK sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From:
"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c9400200b6f3027-
31d1d958
To: <sip:2403649311@192.168.5.253;user=phone>;tag=445624321-1023207995257
Call-ID: 0003e363-0c9406ea-4bd9f593-29482fa1@192.168.5.214
Date: Tue, 04 Jun 2002 21:27:22 GMT
CSeq: 103 ACK
User-Agent: AccessDevice
Max-Forwards:10
Content-Length: 0
```

4.11 Access Device to Cisco BroadWorks Requesting Calling Party Identity Blocking

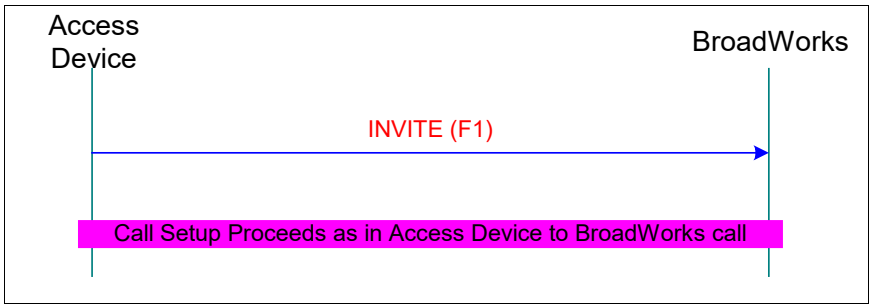

Figure 90 Access Device to Cisco BroadWorks Requesting Calling Party Identity Blocking

4.11.1 F1 – INVITE: Access Device to Cisco BroadWorks

```
INVITE sip:2403645138@192.168.5.253;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
```


©2019 CISCO SYSTEMS, INC.

```
From: "Anonymous" <sip:2403649314@192.168.5.253>;tag=0003e3630c940022398afd1f-
1997cede
To: <sip:2403645138@192.168.5.253;user=phone>
Call-ID: 0003e363-0c940775-422f0d2b-1cf40c2d@192.168.5.214
Date: Wed, 05 Jun 2002 10:55:18 GMT
CSeq: 101 INVITE
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Expires: 180
Content-Type: application/sdp
Max-Forwards:10
Content-Length: 170
Accept: application/sdp

v=0
o=SDP 17148 23635 IN IP4 192.168.5.214
s=SDP
c=IN IP4 192.168.5.214
t=0 0
m=audio 23904 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```

4.12 Access Device to Cisco BroadWorks Call Requesting Calling Party Identity Blocking (RFC 3323/3325)

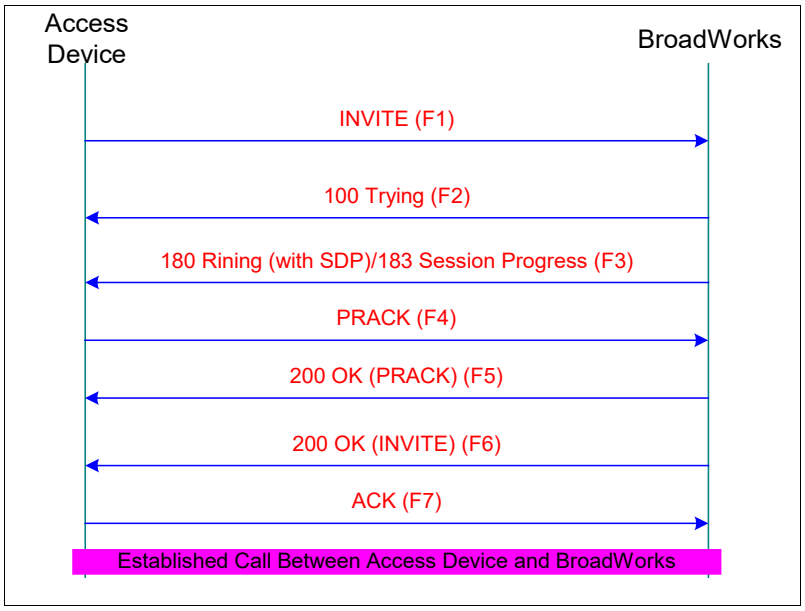

Figure 91 Access Device to Cisco BroadWorks Call Requesting Calling Party Identity Blocking (RFC 3323/3325)

©2019 CISCO SYSTEMS, INC.

4.12.1 F1 – INVITE: Access Device to Cisco BroadWorks

```
INVITE sip:2403645138@192.168.5.253;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=0003e3630c94001a6b35266f-
372ea22b
To: <sip:2403645138@192.168.5.253;user=phone>
Call-ID: 0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
Date: Tue, 04 Jun 2002 19:52:42 GMT
CSeq: 101 INVITE
P-Asserted-Identity: "Bob Smith"<sip:+13015400460@192.168.5.214;user=phone>
Privacy:user;critical;id
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Expires: 180
Supported:100rel,timer
Content-Type: application/sdp
Max-Forwards:10
Content-Length: 170
v=0
o=SDP 26088 15595 IN IP4 192.168.5.214
s=SIP Call
c=IN IP4 192.168.5.214
t=0 0
m=audio 23890 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```

4.12.2 F2 – 100 Trying: Cisco BroadWorks to Access Device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq:101 INVITE
Content-Length:0
```

4.12.3 F3 – 180 Ringing (with or without SDP)/183 Session Progress: Cisco BroadWorks to Access Device

```
SIP/2.0 180 Ringing
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq:101 INVITE
Require:100rel^M
RSeq:1016510814
Content-Length:0
```

4.12.4 F4 – PRACK: Access Device to Cisco BroadWorks

```
PRACK sip: 192.168.5.253;user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq: 102 PRACK
RAck:1016510814 101 INVITE
Max-Forwards:10
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

4.12.5 F5 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq: 102 PRACK
Content-Length: 0
```

4.12.6 F6 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq:101 INVITE
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Contact:<sip:192.168.5.253:5060>
Content-Type:application/sdp
Content-Length:158

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 192.168.5.215
t=0 0
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.12.7 F7 – ACK: Access Device to Cisco BroadWorks

```
ACK sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From: "2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To: <sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID: 0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
Date: Tue, 04 Jun 2002 19:52:48 GMT
CSeq: 101 ACK
User-Agent: AccessDevice
Max-Forwards:10
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

4.13 Access Device to Cisco BroadWorks Call Requesting Calling Party Identity Blocking (draft-ietf-sip-privacy-03)

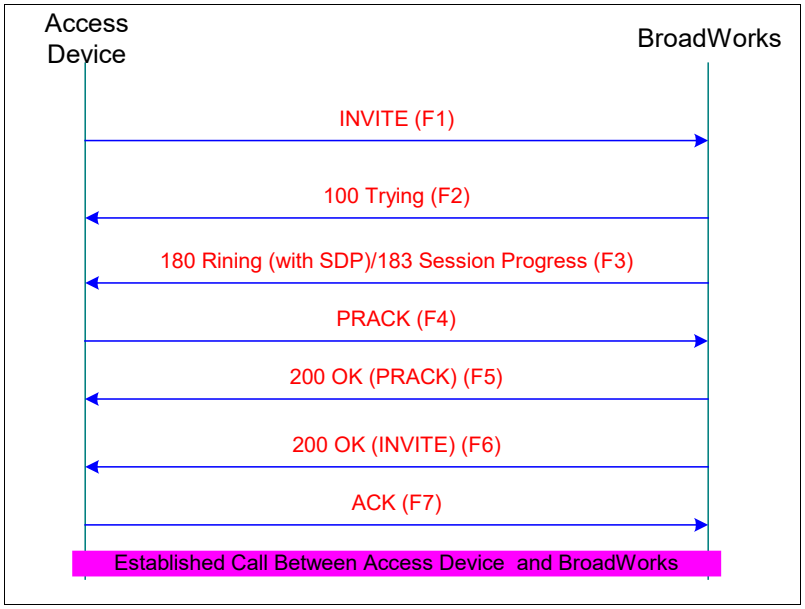

Figure 92 Access Device to Cisco BroadWorks Call Requesting Calling Party Identity Blocking

4.13.1 F1 – INVITE: Access Device to Cisco BroadWorks

```
INVITE sip:2403645138@192.168.5.253;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From"Anonymous"<sip:anonymous@anonymous.invalid>;tag=0003e3630c94001a6b35266f-
372ea22b
To: <sip:2403645138@192.168.5.253;user=phone>
Call-ID: 0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
Date: Tue, 04 Jun 2002 19:52:42 GMT
CSeq: 101 INVITE
RPID-Privacy:party=calling;id-type=subscriber;privacy=full
Remote-Party-ID:"Bob Smith"<sip:+13015400460@192.168.5.214>;party=calling;id-
type=subscriber;privacy=full;screen=yes
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Expires: 180
Supported:100rel,timer
Content-Type: application/sdp
Max-Forwards:10
Content-Length: 170
v=0
o=SDP 26088 15595 IN IP4 192.168.5.214
s=SIP Call
c=IN IP4 192.168.5.214
t=0 0
m=audio 23890 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```

4.13.2 F2 – 100 Trying: Cisco BroadWorks to Access Device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
```


©2019 CISCO SYSTEMS, INC.

```
CSeq:101 INVITE
Content-Length:0
```

4.13.3 F3 – 180 Ringing (with or without SDP)/183 Session Progress: Cisco BroadWorks to Access Device

```
SIP/2.0 180 Ringing
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq:101 INVITE
Require:100rel^M
RSeq:1016510814
Content-Length:0
```

4.13.4 F4 – PRACK: Access Device to Cisco BroadWorks

```
PRACK sip: 192.168.5.253;user=phone SIP/2.0
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq: 102 PRACK
RAck:1016510814 101 INVITE
Max-Forwards:10
Content-Length: 0
```

4.13.5 F5 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq: 102 PRACK
Content-Length: 0
```

4.13.6 F6 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To:<sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID:0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
CSeq:101 INVITE
Content-Type:application/sdp
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Contact:<sip:192.168.5.253:5060>
Content-Type:application/sdp
Content-Length:158

v=0
o=BroadWorks 3 1 IN IP4 192.168.2.133
s=-
c=IN IP4 192.168.2.133
t=0 0
```


©2019 CISCO SYSTEMS, INC.

```
m=audio 17382 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.13.7 F7 – ACK: Access Device to Cisco BroadWorks

```
ACK sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From: "2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001a6b35266f-
372ea22b
To: <sip:2403645138@192.168.5.253;user=phone>;tag=55236541-1023202328737
Call-ID: 0003e363-0c9406d6-124f754f-085ca146@192.168.5.214
Date: Tue, 04 Jun 2002 19:52:48 GMT
CSeq: 101 ACK
User-Agent: AccessDevice
Max-Forwards:10
Content-Length: 0
```

4.14 Cisco BroadWorks to Access Device with Calling Party Identity Blocking

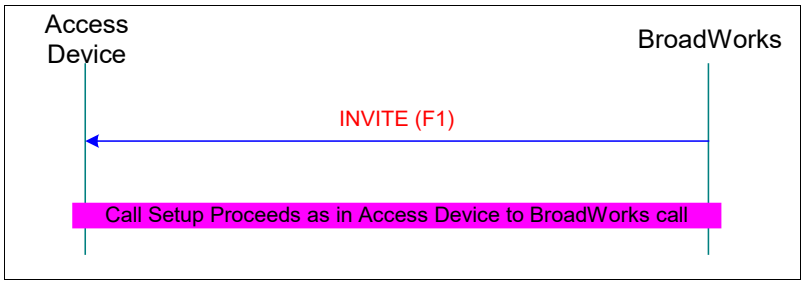

Figure 93 Cisco BroadWorks to Access Device with Calling Party Identity Blocking

4.14.1 F1 – INVITE: Cisco BroadWorks to Access Device

```
INVITE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-192.168.5.214-
V5060-0-527288206-700562235-1023256792932
From:"Anonymous"<sip:anonymous@anonymous.invalid>;tag=700562235-1023256792932
To:"richard ricardo"<sip:2403649314@192.168.5.253;user=phone>
Call-ID:BW015952093205060202 3-628642991177@192.168.5.253
CSeq:527288206 INVITE
Contact:<sip:192.168.5.253:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:158

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 17808 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```


©2019 CISCO SYSTEMS, INC.

4.15 Cisco BroadWorks to Access Device with Priority Alerting Information

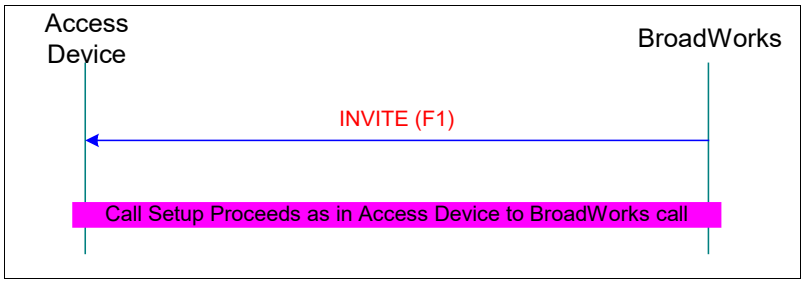

Figure 94 Cisco BroadWorks to Access Device with Priority Alerting Information

4.15.1 F1 – INVITE: Cisco BroadWorks to Access Device

```
INVITE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-192.168.5.214-
V5060-0-527399126-2092450165-1023257014779
From:<sip:2403649311@192.168.5.253;user=phone>;2092450165-1023257014779
To:"richard ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>
Call-ID:BW020334077905060202 3-827529209178@192.168.5.253
CSeq:527399126 INVITE
Contact:<sip:192.168.5.253:5060>
Alert-Info:<http://127.0.0.1/Bellcore-dr2>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:107

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 192.168.5.215
t=0 0
m=audio 16900 RTP/AVP 0
```

4.16 Access Device Initiates Blind Transfer

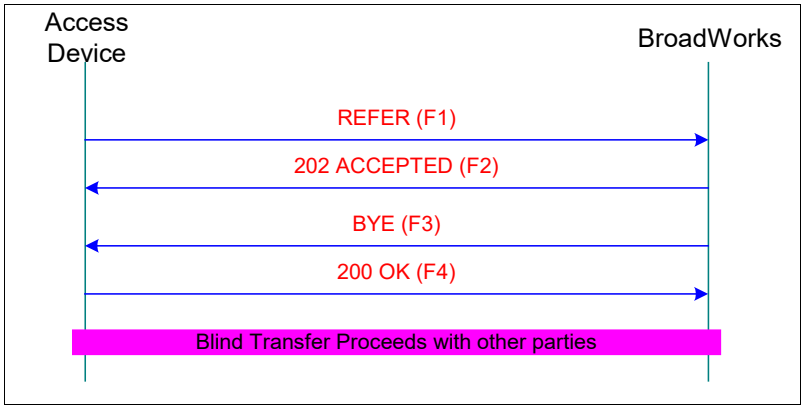

Figure 95 Access Device Initiates Blind Transfer

©2019 CISCO SYSTEMS, INC.

4.16.1 F1 – REFER: Access Device to Cisco BroadWorks

```
REFER sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From: "richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c9400134cb17e63
-24a77a2a
To: <sip:2403649311@192.168.5.253;user=phone>;tag=463641740-1023104752754
Call-ID: BW074552075303060202353435639216 2@192.168.5.253
Date: Mon, 03 Jun 2002 16:46:33 GMT
CSeq: 102 REFER
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Max-Forwards:10
Content-Length: 0
Refer-To: sip:2403645138@192.168.5.253
Referred-By: "richard ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>
```

4.16.2 F2 – 202 Accepted: Cisco BroadWorks to Access Device

```
SIP/2.0 202 Accepted
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c9400134cb17e63
-24a77a2a
To: <sip:2403649311@192.168.5.253;user=phone>;tag=463641740-1023104752754
Call-ID:BW074552075303060202353435639216 2@192.168.5.253
CSeq:102 REFER
Content-Length:0
```

4.16.3 F3 – BYE: Cisco BroadWorks to Access Device

```
BYE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-192.168.5.214-
V5060-0-451268152-463641740-1023104752754
From:<sip:2403649311@192.168.5.253;user=phone>;tag=463641740-1023104752754
To:"richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c9400134cb17e63
-24a77a2a
Call-ID:BW074552075303060202353435639216 2@192.168.5.253
CSeq:451268152 BYE
Max-Forwards:10
Content-Length:0
```

4.16.4 F4 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-451268152-463641740-1023104752754
From: <sip:2403649311@192.168.5.253;user=phone>;tag=463641740-1023104752754
To: "richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c9400134cb17e63
-24a77a2a
Call-ID: BW074552075303060202353435639216 2@192.168.5.253
Date: Mon, 03 Jun 2002 16:46:34 GMT
CSeq: 451268152 BYE
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

4.17 Access Device Initiates Transfer with Consultation

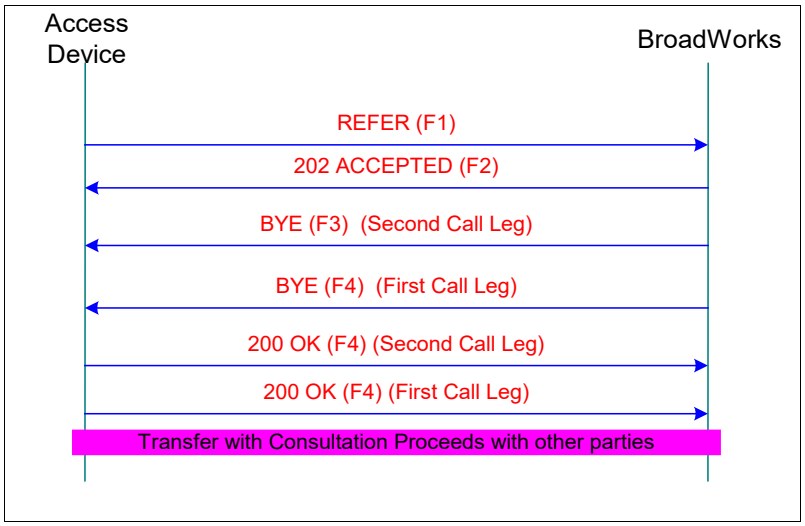

Figure 96 Access Device Initiates Transfer with Consultation

4.17.1 F1 – REFER: Access Device to Cisco BroadWorks

```
REFER sip:192.168.5.253:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.5.214:5060
From: "richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c940014177c792d
-4ca7ec8c
To: <sip:2403649311@192.168.5.253;user=phone>;tag=489683483-1023104804872
Call-ID: BW074644087103060 2023-327253813164@192.168.5.253
Date: Mon, 03 Jun 2002 16:47:37 GMT
CSeq: 102 REFER
User-Agent: AccessDevice
Contact: sip:2403649314@192.168.5.214:5060
Max-Forwards:10
Content-Length: 0
Refer-To: <sip:2403645138@192.168.5.253;user=phone?Replaces=0003e363-0c9405bd-
3004dff7-04942c99%40192.168.5.214%3Bto-tag%3D1408776144-1023104817175%3Bfrom-
tag%3D0003e3630c94001543a82030-63099116>
Referred-By: "richard ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>
```

4.17.2 F2 – 202 Accepted: Cisco BroadWorks to Access Device

```
SIP/2.0 202 Accepted
Via:SIP/2.0/UDP 192.168.5.214:5060
From:"richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c940014177c792d
-4ca7ec8c
To:<sip:2403649311@192.168.5.253;user=phone>;tag=489683483-1023104804872
Call-ID:BW074644087103060 2023-327253813164@192.168.5.253
CSeq:102 REFER
Content-Length:0
```

4.17.3 F3 – BYE: Cisco BroadWorks to Access Device

```
BYE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-192.168.5.214-
V5060-0-451294211-489683483-1023104804872
From: <sip:2403649311@192.168.5.253;user=phone>;tag=489683483-1023104804872
```


©2019 CISCO SYSTEMS, INC.

```
To:"richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c940014177c792d
-4ca7ec8c
Call-ID:BW074644087103060 2023-327253813164@192.168.5.253
CSeq:451294211 BYE
Max-Forwards:10
Content-Length:0
```

4.17.4 F4 – BYE: Cisco BroadWorks to Access Device

```
BYE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-192.168.5.214-
V5060-0-451300184-1408776144-1023104817175
From:<sip:2403645138@192.168.5.253;user=phone>;tag=1408776144-1023104817175
To:"2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001543a82030-63099116
Call-ID:0003e363-0c9405bd-3004dff7-04942c99@192.168.5.214
CSeq:451300184 BYE
Max-Forwards:10
Content-Length:0
```

4.17.5 F5 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-451294211-489683483-1023104804872
From: <sip:2403649311@192.168.5.253;user=phone>;tag=489683483-1023104804872
To: "richard
ricardo"<sip:2403649314@192.168.5.214:5060;user=phone>;tag=0003e3630c940014177c792d
-4ca7ec8c
Call-ID: BW074644087103060 2023-327253813164@192.168.5.253
Date: Mon, 03 Jun 2002 16:47:38 GMT
CSeq: 451294211 BYE
Content-Length: 0
```

4.17.6 F6 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.5.214-V5060-0-451300184-1408776144-1023104817175
From: <sip:2403645138@192.168.5.253;user=phone>;tag=1408776144-1023104817175
To: "2403649314"<sip:2403649314@192.168.5.253>;tag=0003e3630c94001543a82030-
63099116
Call-ID: 0003e363-0c9405bd-3004dff7-04942c99@192.168.5.214
Date: Mon, 03 Jun 2002 16:47:38 GMT
CSeq: 451300184 BYE
Content-Length: 0
```

4.18 Cisco BroadWorks Sends Message Waiting Indication to Access Device

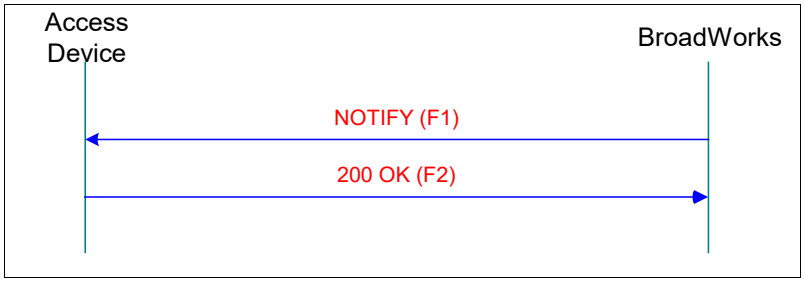

Figure 97 Cisco BroadWorks Sends Message Waiting Indication to Access Device

©2019 CISCO SYSTEMS, INC.

4.18.1 F1 – NOTIFY: Cisco BroadWorks to Access Device

```
NOTIFY sip:2405551015@192.168.242.205 SIP/2.0
Via:SIP/2.0/UDP 192.168.6.20;branch=z9hG4bKBroadWorks.-1su2iau-192.168.242.
205V5060-0-624825602-1344589726-1096466311684
From:<sip:192.168.6.20>;tag=1344589726-1096466311684
To:<sip:2405551015@192.168.242.205>
Call-ID:BW095831684290904-307207238@192.168.6.20
CSeq:624825602 NOTIFY
Contact:<sip:intas.broadworks.net>
Event:message-summary
Subscription-State:terminated
Max-Forwards:10
Content-Type:application/simple-message-summary
Content-Length:43

Messages-Waiting: yes
voice-message: 2/0
```

4.18.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.6.20;branch=z9hG4bKBroadWorks.-1su2iau-192.168.242.
205V5060-0-624825602-1344589726-1096466311684
From: <sip:192.168.6.20>;tag=1344589726-1096466311684
To: <sip:2405551015@192.168.242.205>
Call-ID: BW095831684290904-307207238@192.168.6.20
Date: Mon, 03 Jun 2002 16:46:05 GMT
CSeq: 624825602 NOTIFY
Content-Length: 0
```

4.19 Cisco BroadWorks to Access Device Call with Redirection (Diversion), Unconditional Call Forwarding

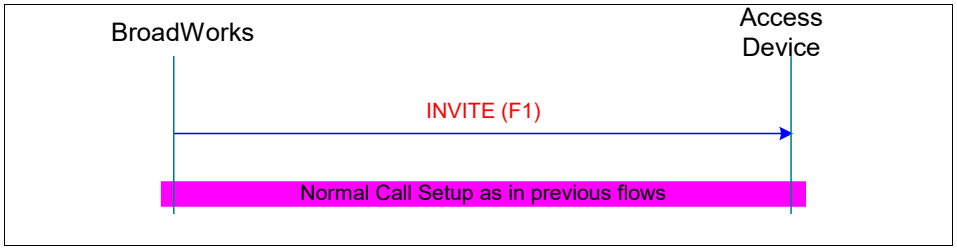

Figure 98 Cisco BroadWorks to Access Device Call with Redirection (Diversion), Unconditional Call Forwarding

4.19.1 F1 – INVITE: Cisco BroadWorks to Access Device

```
INVITE sip:2403649314@192.168.5.214:5060 SIP/2.0
Via:SIP/2.0/UDP 192.168.5.253:5060;branch=z9hG4bKBroadWorks.-1su2iau-192.168.5.214-
V5060-0-501003454-1569407939-1023204223358
From:<sip:2403645138@192.168.2.133;user=phone>;tag=1569407939-1023204223358
To:"richard ricardo"<sip:2403649314@192.168.5.253;user=phone>
Call-ID:BW112343035704060202316497252421 72@192.168.5.253
CSeq:501003454 INVITE
Contact:<sip:192.168.5.253:5060>
Diversion:"Bob
Smith"<sip:+13015400460@applicationserver.broadsoft.com>;reason=unconditional;count
er=1
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER
Supported:100rel,timer
Accept:application/sdp
Max-Forwards:10
```


©2019 CISCO SYSTEMS, INC.

```
Content-Type:application/sdp
Content-Length:158

v=0
o=BroadWorks 3 1 IN IP4 192.168.5.215
s=-
c=IN IP4 192.168.5.215
t=0 0
m=audio 18054 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

4.20 Cisco BroadWorks Informs Access Device to Play Call-Waiting Tone

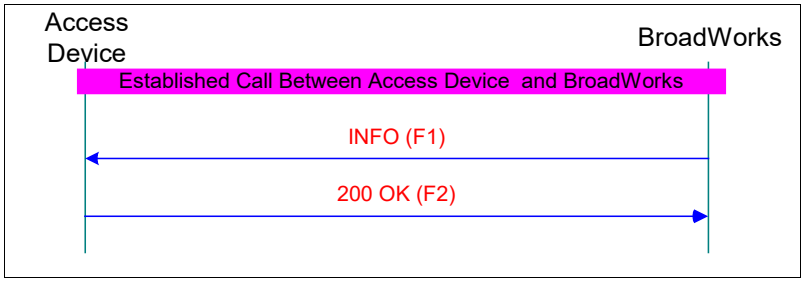

Figure 99 Cisco BroadWorks Informs Access Device to Play Call-Waiting Tone

4.20.1 F1 – INFO: Cisco BroadWorks to Access Device

```
INFO sip:3010050303@10.254.25.1:23072 SIP/2.0
Via:SIP/2.0/UDP 10.0.1.141:5060;branch=z9hG4bKBroadWorks.-1su2iau-
10.254.25.1V23072-0-671332521-931769837-1049314685266
From:"3010050301 NSS"<sip:0301@10.0.1.141;user=phone>;tag=931769837-1049314685266
To:"3010050303
NSS"<sip:3010050303@10.0.1.141;user=phone>;tag=00000000000060E3000C3DB2
Call-ID:BW1518050266020403014-9673143922264 6@10.0.1.141
CSeq:671332521 INFO
Content-Length:28
Max-Forwards:10
Content-Type:application/broadsoft

play tone CallWaitingTone1
Calling-Name:"Rod Smith"
Calling-Number:2403645137
```

4.20.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.0.1.141:5060;branch=z9hG4bKBroadWorks.-1su2iau-
10.254.25.1V23072-0-671332521-931769837-1049314685266
Call-ID:BW1518050266020403014-9673143922264 6@10.0.1.141
CSeq:671332521 INFO
From:"3010050301 NSS"<sip:0301@10.0.1.141;User=phone>;tag=931769837-1049314685266
To:"3010050303
NSS"<sip:3010050303@10.0.1.141;User=phone>;tag=00000000000060E3000C3DB2
Server:1.2
Supported:100rel,timer
Content-Length:0
```


©2019 CISCO SYSTEMS, INC.

4.21 Cisco BroadWorks Informs Access Device to Stop Call-Waiting Tone

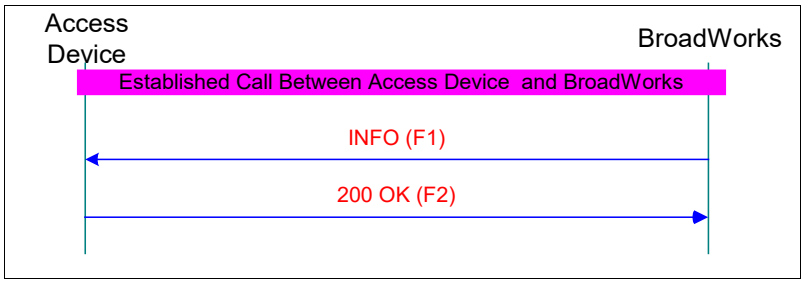

Figure 100 Cisco BroadWorks Informs Access Device to Stop Call-Waiting Tone

4.21.1 F1 – INFO: Cisco BroadWorks to Access Device

```
INFO sip:3010050303@10.254.25.1:23072 SIP/2.0
Via:SIP/2.0/UDP 10.0.1.141:5060;branch=z9hG4bKBroadWorks.-1su2iau-
10.254.25.1V23072-0-671332521-931769837-1049314685266
From:"3010050301 NSS"<sip:0301@10.0.1.141;user=phone>;tag=931769837-1049314685266
To:"3010050303
NSS"<sip:3010050303@10.0.1.141;user=phone>;tag=00000000000060E3000C3DB2
Call-ID:BW151805026602040301 4-96731439222646@10.0.1.141
CSeq:671332521 INFO
Content-Length:28
Max-Forwards:10
Content-Type:application/broadsoft

Stop CallWaitingTone
```

4.21.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.0.1.141:5060;branch=z9hG4bKBroadWorks.-1su2iau-
10.254.25.1V23072-0-671332521-931769837-1049314685266
Call-ID:BW151805026602040301 4-96731439222646@10.0.1.141
CSeq:671332521 INFO
From:"3010050301 NSS"<sip:0301@10.0.1.141;User=phone>;tag=931769837-1049314685266
To:"3010050303
NSS"<sip:3010050303@10.0.1.141;User=phone>;tag=00000000000060E3000C3DB2
Server:1.2
Supported:100rel,timer
Content-Length:0
```

4.22 Access Device Informs Cisco BroadWorks the User Pressed Flash Hook

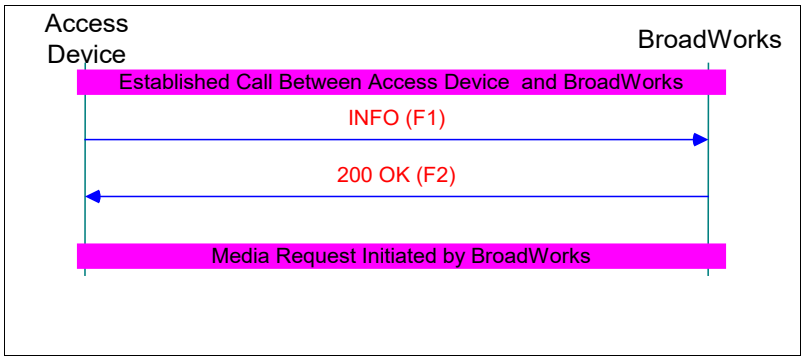

Figure 101 Access Device Informs Cisco BroadWorks the User Pressed Flash Hook

©2019 CISCO SYSTEMS, INC.

4.22.1 F1 – INFO: Access Device to Cisco BroadWorks

```
INFO sip:10.0.1.141:5060 SIP/2.0
Content-Type:application/broadsoft
Call-ID:BW1518050266020403014-9673143922646@10.0.1.141
CSeq:1 INFO
From:"3010050303
NSS"<sip:3010050303@10.0.1.141;User=phone>;tag=00000000000060E3000C3DB2
To:"3010050301 NSS"<sip:0301@10.0.1.141;User=phone>;tag=931769837-1049314685266
Via:SIP/2.0/UDP 10.254.25.1:23072
User-Agent:AmethystUAv0.0.0
Supported:100rel,timer
Max-Forwards:10
Content-Length:17

event flashhook
```

4.22.2 F2 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 10.254.25.1:23072
From:"3010050303
NSS"<sip:3010050303@10.0.1.141;user=phone>;tag=00000000000060E3000C3DB2
To:"3010050301 NSS"<sip:0301@10.0.1.141;user=phone>;tag=931769837-1049314685266
Call-ID:BW1518050266020403014-9673143922646@10.0.1.141
CSeq:1 INFO
Content-Length:0
```

4.23 Access Device to Cisco BroadWorks Subscription (Generic-Event Event Package Example)

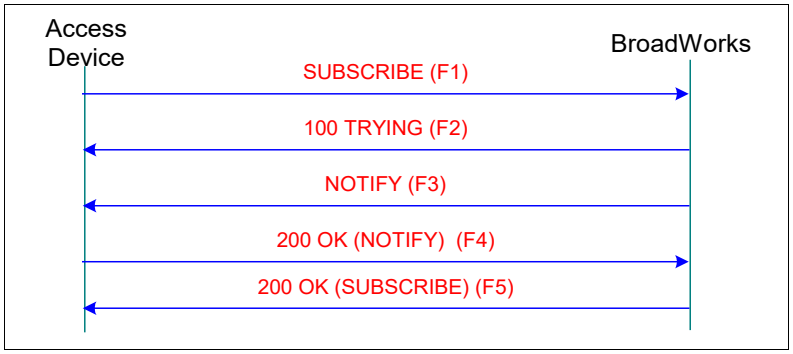

Figure 102 Access Device to Cisco BroadWorks Subscription

4.23.1 F1 – SUBSCRIBE: Access Device to Cisco BroadWorks

```
SUBSCRIBE sip:userg1@mtlasdev5 SIP/2.0
Via: SIP/2.0/UDP 192.168.8.243:11179
From: "userg2" <sip:userg2@mtlasdev5>;tag=260bd5dd-943c-41b0-90eb-facc3093dde3
To: <sip:userg1@mtlasdev5>
Call-ID: 56fff95c-1e44-4fe7-9d2c-c2e627de944b@192.168.8.243
CSeq: 1 SUBSCRIBE
Contact: <sip:192.168.8.243:11179>
User-Agent: Windows RTC/1.0
Expires: 1800
Max-Forwards:10
Content-Length: 0
```


©2019 CISCO SYSTEMS, INC.

4.23.2 F2 – 100 Trying: Cisco BroadWorks to Access Device

```
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.168.8.243:11179
From:"userg2"<sip:userg2@mtlasdev5>;tag=260bd5dd-943c-41b0-90eb-facc3093dde3
To:<sip:userg1@mtlasdev5>
Call-ID:56fff95c-1e44-4fe7-9d2c-c2e627de944b@192.168.8.243
CSeq:1 SUBSCRIBE
Content-Length:0
```

4.23.3 F3 – NOTIFY: Cisco BroadWorks to Access Device

```
NOTIFY sip:192.168.8.243:11179 SIP/2.0
Via: SIP/2.0/UDP 192.168.8.18:7806
From: <sip:userg1@mtlasdev5>;tag=c4cf834d-0e02-4cbe-953b-d93474c4aeef
To: "userg2"<sip:userg2@mtlasdev5>;tag=260bd5dd-943c-41b0-90eb-facc3093dde3
Call-ID: 56fff95c-1e44-4fe7-9d2c-c2e627de944b@192.168.8.243
CSeq: 1 NOTIFY
Contact: <sip:192.168.8.18:7806>
Max-Forwards:10
User-Agent: Windows RTC/1.0
Content-Type: application/xpidf+xml
Content-Length: 343
<?xml version="1.0"?>
<!DOCTYPE presence
PUBLIC "-//IETF//DTD RFCxxxx XPIDF 1.0//EN" "xpidf.dtd">
<presence>
<presentity uri="sip:userg2@mtlasdev5;method=SUBSCRIBE" />
<atom id="1016">
<address uri="sip:192.168.8.18:7806;user=ip" priority="0.800000">
<status status="open" />
<msnsubstatus substatus="online" />
</address>
</atom>
</presence>
```

4.23.4 F4 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.8.18:7806
From: <sip:userg1@mtlasdev5>;tag=c4cf834d-0e02-4cbe-953b-d93474c4aeef
To: "userg2" <sip:userg2@mtlasdev5>;tag=260bd5dd-943c-41b0-90eb-facc3093dde3
Call-ID: 56fff95c-1e44-4fe7-9d2c-c2e627de944b@192.168.8.243
CSeq: 1 NOTIFY
User-Agent: Windows RTC/1.0
Content-Length: 0
```

4.23.5 F5 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.168.8.243:11179
From:"userg2"<sip:userg2@mtlasdev5>;tag=260bd5dd-943c-41b0-90eb-facc3093dde3
To:<sip:userg1@mtlasdev5>;tag=c4cf834d-0e02-4cbe-953b-d93474c4aeef
Call-ID:56fff95c-1e44-4fe7-9d2c-c2e627de944b@192.168.8.243
CSeq:1 SUBSCRIBE
Contact:<sip:192.168.8.18:7806>
User-Agent:Windows RTC/1.0
Expires:1799
Content-Length:0
```


©2019 CISCO SYSTEMS, INC.

4.24 Cisco BroadWorks to Access Device Subscription (Generic-Event Event Package Example)

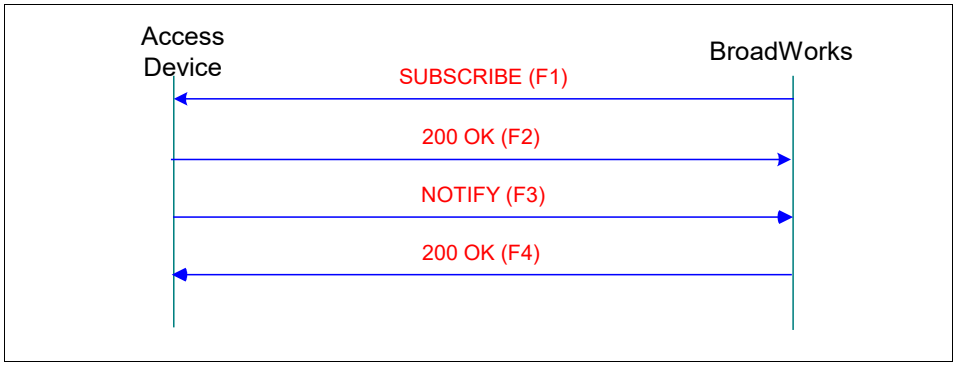

Figure 103 Cisco BroadWorks to Access Device Subscription

4.24.1 F1 – SUBSCRIBE: Cisco BroadWorks to Access Device

```
SUBSCRIBE sip:userg2@mtlasdev5 SIP/2.0
Via:SIP/2.0/UDP 192.168.8.52:5066;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.8.243V11179-0-1-1f00b14e-e08d-4f48-95bb-0e5400df2797
Via:SIP/2.0/UDP 192.168.8.18:7806
From:"userg1"<sip:userg1@mtlasdev5>;tag=1f00b14e-e08d-4f48-95bb-0e5400df2797
To:<sip:userg2@mtlasdev5>
Call-ID:8500cb16-38a5-4d06-8914-04d028ff742f@192.168.8.18
CSeq:1 SUBSCRIBE
Contact:<sip:192.168.8.18:7806>
Max-Forwards:10
User-Agent:Windows RTC/1.0
Expires:1799
Content-Length:0
```

4.24.2 F2 – 200 OK: Access Device to Cisco BroadWorks

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.8.52:5066;branch=z9hG4bKBroadWorks.-1su2iau-
192.168.8.243V11179-0-1-1f00b14e-e08d-4f48-95bb-0e5400df2797
Via: SIP/2.0/UDP 192.168.8.18:7806
From: "userg1"<sip:userg1@mtlasdev5>;tag=1f00b14e-e08d-4f48-95bb-0e5400df2797
To: <sip:userg2@mtlasdev5>;tag=c7b0ccb8-f33f-4e20-aefa-bd5bc06f81ae
Call-ID: 8500cb16-38a5-4d06-8914-04d028ff742f@192.168.8.18
CSeq: 1 SUBSCRIBE
Contact: <sip:192.168.8.243:11179>
User-Agent: Windows RTC/1.0
Expires: 1799
Content-Length: 0
```

4.24.3 F3 – NOTIFY: Access Device to Cisco BroadWorks

```
NOTIFY sip:192.168.8.18:7806 SIP/2.0
Via: SIP/2.0/UDP 192.168.8.243:11179
From: <sip:userg2@mtlasdev5>;tag=c7b0ccb8-f33f-4e20-aefa-bd5bc06f81ae
To: "userg1"<sip:userg1@mtlasdev5>;tag=1f00b14e-e08d-4f48-95bb-0e5400df2797
Call-ID: 8500cb16-38a5-4d06-8914-04d028ff742f@192.168.8.18
CSeq: 1 NOTIFY
Contact: <sip:192.168.8.243:11179>
User-Agent: Windows RTC/1.0
Content-Type: application/xpidf+xml
Max-Forwards:10
Content-Length: 345
```


©2019 CISCO SYSTEMS, INC.

```
<?xml version="1.0"?>
<!DOCTYPE presence
PUBLIC "-//IETF//DTD RFCxxxx XPIDF 1.0//EN" "xpidf.dtd">
<presence>
<presentity uri="sip:userg1@mtlasdev5;method=SUBSCRIBE" />
<atom id="2753">
<address uri="sip:192.168.8.243:11179;user=ip" priority="0.800000">
<status status="open" />
<msnsubstatus substatus="online" />
</address>
</atom>
</presence>
```

4.24.4 F4 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.8.243:11179
From: <sip:userg2@mtlasdev5>;tag=c7b0ccb8-f33f-4e20-aefa-bd5bc06f81ae
To: "userg1" <sip:userg1@mtlasdev5>;tag=1f00b14e-e08d-4f48-95bb-0e5400df2797
Call-ID: 8500cb16-38a5-4d06-8914-04d028ff742f@192.168.8.18
CSeq: 1 NOTIFY
User-Agent: Windows RTC/1.0
Content-Length: 0
```

4.25 Access Device to Cisco BroadWorks SMS

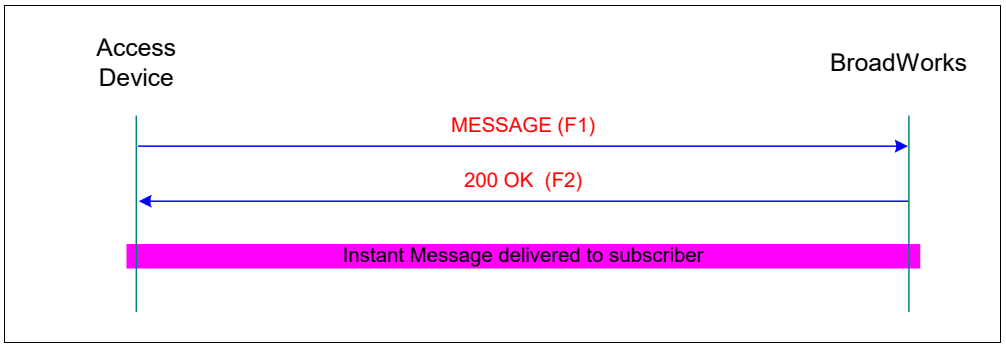

Figure 104 Access Device to Cisco BroadWorks SMS

4.25.1 F1 – MESSAGE: Access Device to Cisco BroadWorks

```
MESSAGE sip:7035553001@initech.test SIP/2.0
Via: SIP/2.0/UDP 10.16.145.5;branch=z9hG4bK-2700257532
From: <sip:1008@initech.test>;tag=8136091812
To: <sip:7035553001@initech.test>
Call-ID: 5296166829
CSeq: 1 MESSAGE
Max-Forwards: 70
Content-Type: text/plain;charset=utf-8
Content-Length: 10

Hi, Mom!
```


©2019 CISCO SYSTEMS, INC.

4.25.2 F2 – 200 OK: Cisco BroadWorks to Access Device

```
SIP/2.0 202 Accepted
Via:SIP/2.0/UDP 10.16.145.5;branch=z9hG4bK-2700257532
From:<sip:1008@initech.test>;tag=8136091812
To:<sip:7035553001@initech.test>;tag=1114072478-1285957444276
Call-ID:5296166829
CSeq:1 MESSAGE
Content-Length:0
```

4.26 Network Server Redirection for REGISTER

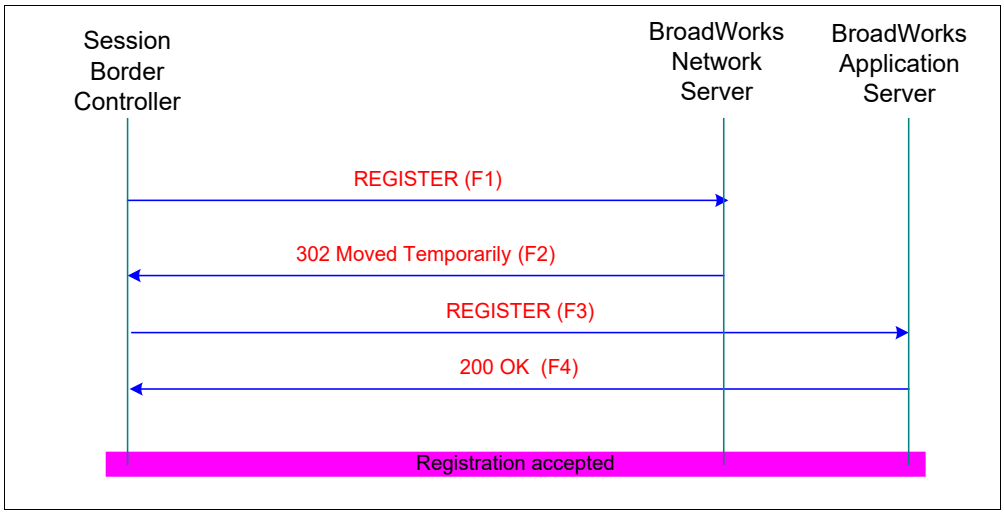

Figure 105 Network Server Redirection for Register

4.26.1 F1 – REGISTER: Session Border Controller to Network Server

```
REGISTER sip:ns.domain.net SIP/2.0
Via:SIP/2.0/UDP sbc.domain.net; branch=z9hG4bK649538DA591673
From:"test phone"<sip:user1@domain.net>;tag=2245653809
To:"test phone"<sip:user1@domain.net>
Contact:"test phone"<sip:user1@mydomain.net>
Call-ID:CF2FF8418F5E406F94A516C2F5708117@mydomain.net
CSeq:50417 REGISTER
Expires:1800
Max-Forwards:70
Content-Length:0
```

4.26.2 F2 – 302 Moved Temporarily: Network Server to Session Border Controller

```
SIP/2.0 302 Moved temporarily
Via:SIP/2.0/UDP sbc.domain.net; branch=z9hG4bK649538DA591673
From:"test phone"<sip:user1@domain.net>;tag=2245653809
To:"test phone"<sip:user1@domain.net>
Call-ID:CF2FF8418F5E406F94A516C2F5708117@mydomain.net
CSeq:50417 REGISTER
Contact:<sip:as1.domain.net;user=phone>;q=0.5,<sip:as2.domain.net;user=phone>;q=0.5
```


©2019 CISCO SYSTEMS, INC.

4.26.3 F3 – REGISTER: Session Border Controller to Application Server

```
REGISTER sip:as1.domain.net SIP/2.0
Via:SIP/2.0/UDP sbc.domain.net; branch=z9hG4bK649538DA591674
From:"test phone"<sip:user1@domain.net>;tag=2245653810
To:"test phone"<sip:user1@domain.net>
Contact:"test phone"<sip:user1@mydomain.net>
Call-ID:CF2FF8418F5E406F94A516C2F5708118@mydomain.net
CSeq:50418 REGISTER
Expires:1800
Max-Forwards:70
Content-Length:0
```

4.26.4 F4 – 200 OK: Application Server to Session Border Controller

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP sbc.domain.net; branch=z9hG4bK649538DA591674
From:"test phone"<sip:user1@domain.net>;tag=2245653810
To:"test phone"<sip:user1@domain.net>
Contact:"test phone"<sip:user1@mydomain.net>;q=0.5;expires=1800
Call-ID:CF2FF8418F5E406F94A516C2F5708118@mydomain.net
CSeq:50418 REGISTER
Content-Length:0
```


©2019 CISCO SYSTEMS, INC.

5 Appendix A: SDP Overview

This section is a brief Session Description Protocol (SPD) overview. For a more detailed description of SDP see *RFC 4566* at http://www.ietf.org/rfc/rfc4566.txt. For a more detailed description of SDP usage in SIP, see *RFC 3261 and RFC 3264.*

SIP indicates the use of SDP by setting the entity-header's Content-Type to "application/sdp", and Content-Length to the length of the SDP body. The SDP body starts at the end of the last SIP header; the last SIP header is followed by a blank line (nothing preceding a CRLF); the SDP body starts after this blank line.

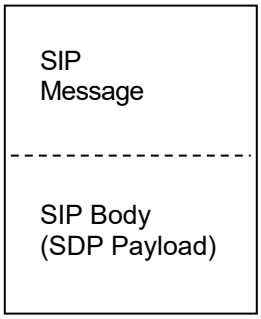

Figure 106 SDP Message

5.1 SDP Sections

There are three main sections in a 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 equal sign.

Some lines in each section are optional, but each line must appear in the order shown in the following section.

5.1.1 Session Description

Session Description applies to the session being setup and all media streams being established. In SIP, only one Session Description is allowed per SIP message, though 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*

©2019 CISCO SYSTEMS, INC.

c = (connection information is not required 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*

©2019 CISCO SYSTEMS, INC.

SIP SDP Example:

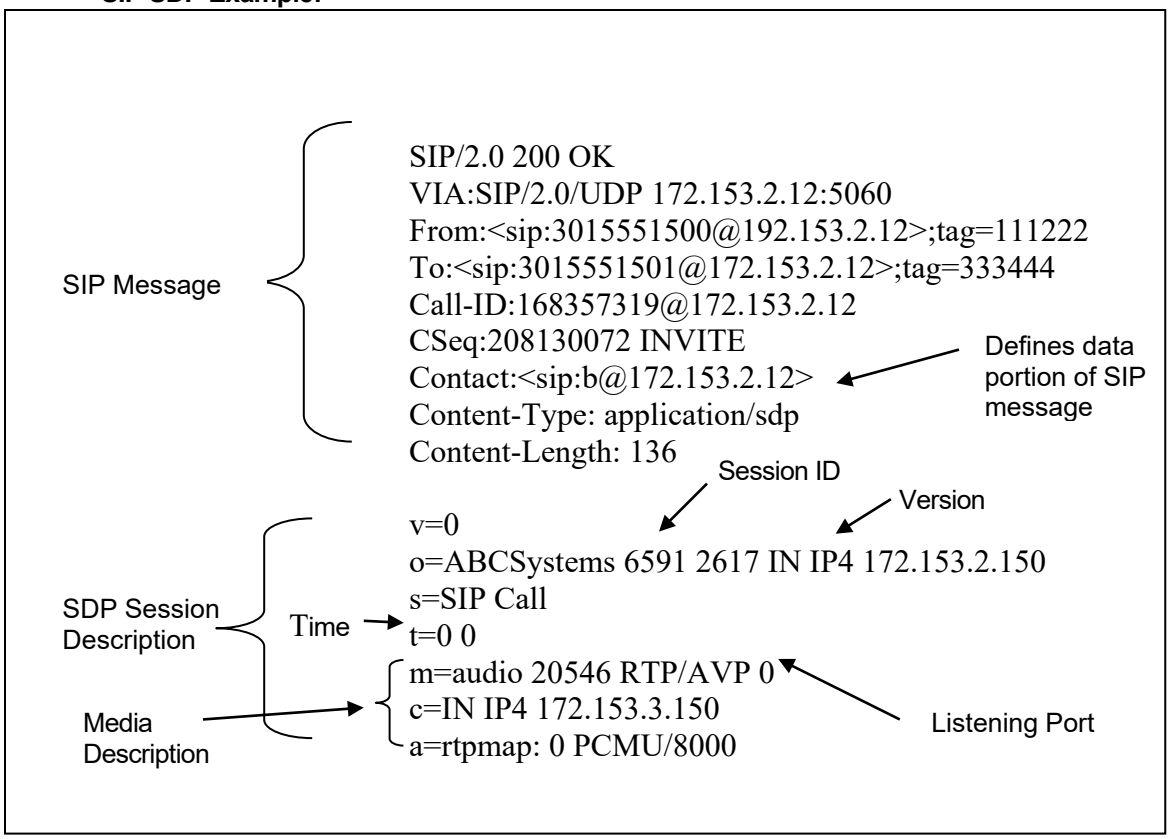

Figure 107 SDP Example

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 to 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.

For 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;branch=z9hG4bK-123
                  From: sip:<A@192.168.2.12>; tag=333999
                  To: <sip:B@192.168.2.12;user=phone>
                  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
```


©2019 CISCO SYSTEMS, INC.

```
                    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;branch=z9hG4bK-123
                    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
```

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. In addition, to remove a media stream from a call, either side sends a re-INVITE and sets the port it wants to remove to zero.

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 sync up, 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>
```


©2019 CISCO SYSTEMS, INC.

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* or *a=sendonly*, and increment the version field of the “o” line.

©2019 CISCO SYSTEMS, INC.

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, Internet Engineering Task Force, December 2004. Available from http://www.ietf.org/.

[4] 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/.

[5] 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/.

[6] 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.

[7] 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.

[8] Levy, S., Byerly, B., Yang, J. R., "Diversion Indication in SIP", draft-levy-sip-diversion-08, August 25, 2004.

[9] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, Internet Engineering Task Force, September 2002. Available from http://www.ietf.org/.

[10] 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/.

[11] Donovan, S., Rosenberg J., "Session Timers in the Session Initiation Protocol (SIP)", RFC 4028, Internet Engineering Task Force, April 2005. Available from http://www.ietf.org/.

[12] 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/.

[13] Mahy, R., Gurbani, V., Tate, B., "Connection Reuse in the Session Initiation Protocol (SIP)", RFC 5923, Internet Engineering Task Force, August 21, 2006. Available from http://www.ietf.org/.

[14] 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, Internet Engineering Task Force, April 2004. Available from http://www.ietf.org/

[15] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, Internet Engineering Task Force, June 2002. Available from http://www.ietf.org/.

©2019 CISCO SYSTEMS, INC.

[16] Donovan, S., "The SIP INFO Method", RFC 2976, Internet Engineering Task Force, October 2000. Available from http://www.ietf.org/.

[17] 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/.

[18] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, Internet Engineering Task Force, April 2003. Available from http://www.ietf.org/.

[19] Mahy, R., Biggs, B, Dean, R., "The Session Initiation Protocol (SIP) Replaces Header", RFC 3891, Internet Engineering Task Force, September 2004. Available from http://www.ietf.org/.

[20] Sparks, R., "The Session Initiation Protocol (SIP) Referred-By Mechanism", RFC 3892, Internet Engineering Task Force, September 2004. Available from http://www.ietf.org/.

[21] 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/.

[22] 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/.

[23] ITU-T T.38 *Procedures for real-time Group 3 Facsimile communication over IP networks*, September 2005.

[24] International Telecommunication Union, August 2003, *SIP Support for Real-time Fax: Call Flow Examples and Best Current Practices*. Available from http://www3.ietf.org/.

[25] International Telecommunication Union, Recommendation T.38 (2004), *Procedures for real-time Group 3 facsimile communication over IP networks.* Available from http://www.itu.int/.

[26] Buckley, R., Venable, D., McIntyre, L., Parsons, G., and Rafferty, J. "File Format for Internet Fax", RFC 3949, Internet Engineering Task Force, February 2005. Available from http://www.ietf.org/.

[27] International Telecommunication Union, Recommendation T.30, July, 1996, *Procedures for document facsimile transmission in the general switched telephone network.* Available from http://www.vsi.ru/library/ITU-T/.

[28] Handley, M., Jacobson, V., "SDP: Session Description Protocol", RFC 2327, Internet Engineering Task Force, April 1998. Available from http://www.ietf.org/.

[29] 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/.

[30] 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/.

[31] 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/.

[32] 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/.

©2019 CISCO SYSTEMS, INC.

[33] 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/.

[34] 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/.

[35] 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/.

[36] 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/.

[37] 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/.

[38] 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+)", RFC 4629, Internet Engineering Task Force, January 2007. Available from http://www.ietf.org/.

[39] Rosenberg, J., Schulzrinne, H., Mahy, R., "An INVITE Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", RFC 4235, Internet Engineering Task Force, November 2005. Available from http://www.ietf.org/.

[40] Johnston, A., Levin, O., "Session Initiation Protocol (SIP) Call Control – Conferencing for User Agents", RFC 4579, Internet Engineering Task Force, August 2006. Available from http://www.ietf.org/.

[41] Rosenberg, J., Schulzrinne, H., Levin, O., "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, Internet Engineering Task Force, August 2006. Available from http://www.ietf.org/.

[42] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, Internet Engineering Task Force, February 2006. Available from http://www.ietf.org/.

[43] BroadSoft, Inc. 2016. *BroadWorks SIP Access Side Extensions Interface Specification, Release 22.0*. Available from BroadSoft at xchange.broadsoft.com.

[44] 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/.

[45] 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/.

[46] BroadSoft, Inc. 2016. *BroadWorks Treatment Guide, Release 22.0*. Available from BroadSoft at xchange.broadsoft.com.

[47] 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/.

[48] 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/.

©2019 CISCO SYSTEMS, INC.

[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] 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/.

[51] 3rd Generation Partnership Project (3GPP). 2007. 3GPP TS 24.647 v8.0.0 Advice of Charge (AoC). Available from http://www.3gpp.org/.

[52] 3rd Generation Partnership Project (3GPP). 2007. 3GPP TS 24.647 v8.0.0 Advice of Charge (AoC). Available from http://www.3gpp.org/.

[53] Mahy, R., Petrie, D., "The Session Initiation Protocol (SIP) Join" Header", RFC 3911, Internet Engineering Task Force, October 2004. Available from http://www.ietf.org/.

[54] Levy, S., Mohali, M., "Diversion Indication in SIP", RFC 5806, Internet Engineering Task Force, March 2010. Available from http://www.ietf.org/.

[55] 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/.

[56] 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/.

[57] Handley, M., Jacobson, V., Jacobson, V., "SDP: Session Description Protocol", RFC 4566, Internet Engineering Task Force, July 2006. Available from http://www.ietf.org/.

[58] 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/.

[59] 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/.

[60] 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.

[61] Roach, A.B, "SIP-Specific Event Notification", RFC 6665, Internet Engineering Task Force, July 2012. Available from http://www.ietf.org.

[62] 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.

[63] BroadSoft, Inc. 2016. *BroadWorks Redundancy Guide, Release 22.0*. Available from BroadSoft at xchange.broadsoft.com.

[64] BroadSoft, Inc. 2016. *BroadWorks AS Mode ISC Interface Guide, Release 22.0*. Available from BroadSoft at xchange.broadsoft.com.

[65] BroadSoft, Inc. 2016. *BroadWorks Call Recording Interface Guide, Release 22.0*. Available from BroadSoft at xchange.broadsoft.com.

[66] BroadSoft, Inc. 2018. *BroadWorks SIP Trunking Solution Guide, Release 22.0*. Available from BroadSoft at xchange.broadsoft.com.

[67] BroadSoft, Inc. 2016. *BroadWorks Shared Call Appearance Interface Specification, Release 22.0*. Available from BroadSoft at xchange.broadsoft.com.

©2019 CISCO SYSTEMS, INC.

[68] GSMA. 2015. IR.94 IMS Profile for Conversational Video Service, Version 10.0. Available from http://www.gsma.com/.

©2019 CISCO SYSTEMS, INC.

Acronyms and Abbreviations

ABNF	Augmented Backus-Naur Format
ACM	Address Complete Message
ALTC	Alternate Connectivity
AoC	Advice of Charge
CIF	Common Intermediate Format
CLI	Command Line Interface
CLID	Calling Line Identity
COLP	Connected Line Identification Presentation
CPE	Customer Premises Equipment
CPG	Call Progress Message
DGC	Distributed Group Call
DN	Directory Number
DNS	Domain Name System
DPUBI	Directed Call Pickup Barge-in
DTMF	Dual-tone Multi-frequency
FAC	Feature Access Code
FQDN	Fully Qualified Domain Names
FTP	File Transfer Protocol
FXS	Foreign eXchange Subscriber
GIN	Globally Identifiable Number
IAM	Initial Address Message
ICID	IMS Charging Identity
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IM&P	Instant Messaging and Presence
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISUP	Integrated Services User Part
IVR	Interactive Voice Response
MGCP	Media Gateway Control Protocol
MSCML	Media Server Control Markup Language
NAPTR	Naming Authority Pointer
NAT	Network Address Translation

©2019 CISCO SYSTEMS, INC.

NGN	Next Generation Network
PBX	Private Branch Exchange
PSTN	Public Switched Telephone Network
QCIF	Quarter Common Intermediate Format
RFC	Request for Changes
RTCP	RTP Control Protocol
RTP	Real-Time Transport Protocol
SBC	Session Border Controller
SDP	Session Description Protocol
SHLR	Smart Home Location Register
SIP	Session Initiation Protocol
SMS	Short Message Service
SMSC	Short Message Service Center
SPD	Session Description
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
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
VAD	Voice Activity Detection

©2019 CISCO SYSTEMS, INC.

Index

©2019 CISCO SYSTEMS, INC.

©2019 CISCO SYSTEMS, INC.