PacketCable™

PacketCable Interconnect Guidelines Specification

PKT-SP-IGS-C01-130930

CLOSED

Notice

DISCLAIMER

This document is published by Cable Television Laboratories, Inc. ("CableLabs®").

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various agencies; technological advances; or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein. CableLabs makes no representation or warranty, express or implied, with respect to the completeness, accuracy, or utility of the document or any information or opinion contained in the report. Any use or reliance on the information or opinion is at the risk of the user, and CableLabs shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

This document is not to be construed to suggest that any affiliated company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any cable member to purchase any product whether or not it meets the described characteristics. Nothing contained herein shall be construed to confer any license or right to any intellectual property, whether or not the use of any information herein necessarily utilizes such intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	PKT-SP-IGS-C01-130930			
Document Title:	PacketCable Interconnect Guidelines Specification			
Revision History:	I01 - Released February 28, 2011			
	C01 - Closed September 30, 2013			
Date:	September 30, 2013			
Status:	~~Work in Progress~~	~~Draft~~	~~Issued~~	**Closed**
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

1 SCOPE 1

1.1 Introduction and Purpose 1
1.2 Relationship to Other PacketCable Specifications 2
1.3 Requirements 2

2 REFERENCES 3

2.1 Normative References 3
2.2 Informative References 4
2.3 Reference Acquisition 4

3 TERMS AND DEFINITIONS 5

4 ABBREVIATIONS AND ACRONYMS 6

5 OVERVIEW 7

6 GENERAL PROCEDURES 8

6.1 Extension Negotiation 8
6.2 Public User Identities 8
6.2.1 Identifying the Called User 8
6.2.2 Identifying the Calling User 9
6.3 Trust Domain and Asserted Identities 9
6.4 IPv4/6 Interworking 9
6.5 Fault Isolation and Recovery 9
6.5.1 Interface Failure Detection 9
6.5.2 Congestion Control 10
6.5.3 Session Timer 10
6.5.4 RTP Loopback Test 10
6.6 Media 10

7 CALL FEATURES 12

7.1 Session Establishment 12
7.1.1 SDP Requirements 12
7.1.2 Basic Call Setup 12
7.1.3 Ringback Tone vs. Early Media 12
7.1.4 Early-Media with Multiple Terminating Endpoints 13
7.1.5 Establishing calls using 3PCC 14
7.1.6 Hold 14
7.2 Calling Name and Number Deliver (with Privacy) 15
7.3 Call Forwarding 15
7.3.1 Detecting Call-Forwarding Loops 15
7.4 Call Transfer 16
7.4.1 Call Transfer using REFER/Replaces 16
7.4.2 Call Transfer Using 3PCC 17
7.5 3-Way Conference 17
7.6 Auto Recall/Callback 17
7.6.1 Originating SSP Network Sends INVITE to Target 17
7.6.2 Originating SSP Network Sends SUBSCRIBE to Target 18
7.6.3 Target Sends NOTIFY to Originating SSP Network 18

ANNEX A RESPONSE CODES 19

APPENDIX I ACKNOWLEDGEMENTS.....21

Figures

Figure 1 - Peering Architecture.....7

Tables

Table 1 - Called Public User Identities.....9
Table 2 - Response Codes.....19

This page has been left blank intentionally.

1 SCOPE

1.1 Introduction and Purpose

This specification defines a base set of interconnect procedures to address common interworking issues at the peering interface between two PacketCable networks. It focuses on the interworking procedures required to support basic telephone service, including the following capabilities:

- On-net to on-net calls
- Caller ID with Privacy
- Early media
- Voice, FAX, DTMF-relay
- Local Number Portability
- Call hold/conf/xfer
- Call forwarding
- Auto Recall/Callback
- Fault Isolation and Recovery
- SIP RTP (Session Initiation Protocol, Real-time Transport Protocol) Loopback
- Inter-network keep-alives

Interworking procedures to support the following capabilities are not addressed in this document:

- Calls to/from PSTN (Public Switched Telephone Network)
- Operator calls
- 0+, 0-, busy-line-verify
- Emergency calls
- Transmission loss plan
- Operational capabilities
- Accounting
- Electronic Surveillance
- Quality-of-Service
- Authentication and Security

The interworking procedures described by this specification are generally applicable to any network architecture. Therefore, while the current version of the document focuses on peering PacketCable 1.5 networks, the procedures described here equally apply when one or both of the peering networks are PacketCable 2.0.

1.2 Relationship to Other PacketCable Specifications

This specification defines procedures associated with protocols that are common to other PacketCable specifications. For example, this specification places normative requirements on the SIP interface between two Call Management Servers (CMS), an interface that is fully specified by the CMSS specification [CMSS]. This specification is not meant to contradict or override other PacketCable specifications with which it overlaps. Rather, the intention of this specification is to complement these other PacketCable specifications by defining more detailed procedures to resolve known and specific interworking issues associated with the support of basic telephony services. Vendors should look to the existing PacketCable 1.5 and 2.0 specifications as the de-facto standard in terms of PacketCable compliance.

1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[E.164] ITU-T Recommendation E.164, The international public telecommunication numbering plan, November 2010.

[E-DVA] PacketCable Residential SIP Telephony E-DVA Specification, PKT-SP-RST-E-DVA-I10-121030, October 30, 2012, Cable Television Laboratories, Inc.

[RFC 3261] IETF RFC 3261, SIP: Session Initiation Protocol, June 2002.

[RFC 3264] IETF RFC 3264, An Offer/Answer Model with Session Description Protocol (SDP), June, 2002.

[RFC 3265] IETF RFC 3265, SIP-Specific Event Notification, June 2002.

[RFC 3323] IETF RFC 3323, A Privacy Mechanism for the Session Initiation Protocol (SIP), November 2002.

[RFC 3325] IETF RFC 3325, Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, November 2002.

[RFC 3515] IETF RFC 3515, The Session Initiation Protocol (SIP) Refer Method, April 2003.

[RFC 3551] IETF RFC 3551/STD0065, RTP Profile for Auditor and Video Conferences with Minimal Control, July 2003.

[RFC 3725] IETF RFC 3725, Best Current Practices for Third Party Call Control (3pcc) in SIP, April 2004.

[RFC 3891] IETF RFC 3891, The Session Initiation Protocol (SIP) Replaces Header, September 2004.

[RFC 3966] IETF RFC 3966, The tel URI for Telephone Numbers, December, 2004.

[RFC 4028] IETF RFC 4028, Session Timers in the Session Initiation Protocol, April, 2005.

[RFC 4235] IETF RFC 4235, An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP), November 2005.

[RFC 4244] IETF RFC 4244, An Extension to the Session Initiation Protocol (SIP) for Request History Information, November, 2005.

[RFC 4458] IETF RFC 4458, Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR), April 2006.

[RFC 4566] IETF RFC 4566, SDP: Session Description Protocol, July 2006.

[RFC 4694] IETF RFC 4694, Number Portability Parameters for the "tel" URI, October 2006.

[RFC 4733] IETF RFC 4733, RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals, December 2006.

[RFC 5627] IETF RFC 5627, Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP), October 2009.

[T.38] ITU-T Recommendation T.38, Procedures for Real-Time Group 3 Facsimile Communication over IP Networks, September 2010.

[V.152] ITU-T Recommendation V.152, Procedures for supporting Voice-Band Data over IP Networks, September 2010.

2.2 Informative References

This specification uses the following informative references.

[CMSS] PacketCable CMS to CMS Signaling 1.5 Specification, PKT-SP-CMSS1.5-I07-120412, April 12, 2012, Cable Television Laboratories Inc.

[RFC 3603] IETF RFC 3603, Private Session Initiation Protocol (SIP) Proxy-to-Proxy Extensions for Supporting the PacketCable Distributed Call Signaling Architecture, October 2003.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- Internet Engineering Task Force (IETF) Secretariat, 48377 Fremont Blvd., Suite 117, Fremont, California 94538, USA, Phone: +1-510-492-4080, Fax: +1-510-492-4001, http://www.ietf.org
- Internet Engineering Task Force (IETF), Internet: http://www.ietf.org/
- International Telecommunication Union (ITU), http://www.itu.int/ITU-T/

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Back-to-Back User Agent	A back-to-back user agent (B2BUA) is a logical entity defined in [RFC 3261]. It receives a SIP request and processes it as a user agent server (UAS) up through the SIP protocol layers to the Transaction User (TU) layer, where it is passed via undefined application logic to the TU of a user agent client (UAC). The UAC then generates a request based on the received TU event. Responses received by the UAC are passed to the UAS in the reverse direction. The B2BUA is therefore a concatenation of a UAC and UAS. No explicit definition is defined for the application behavior.
Data Path Border Element	A data path border element (DBE) is located on the administrative border of a domain through which flows the media associated with an inter-domain session. It typically provides media-related functions such as deep packet inspection and modification, media relay, and firewall traversal support. The DBE may be controlled by the SBE.
Media Endpoint	Any entity that terminates an RTP/RTCP stream.
Signaling Path Border Element	A signaling path border element (SBE) is located on the administrative border of a domain through which inter-domain session layer messages will flow. It typically provides signaling functions such as protocol inter-working (for example, H.323 to SIP), identity and topology hiding, and Session Admission Control for a domain.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

3PCC	3rd Party Call Control
AC	Auto-Callback
AR	Auto-Recall
B2BUA	Back-to-Back User Agent
CFBL	Call Forwarding Busy Line
CFDA	Call Forwarding Don't Answer
CFV	Call Forwarding Variable
CMS	Call Management Server
DBE	Data Path Border Element
DSP	Digital Signal Processor
E-DVA	Embedded Digital Voice Adapter
E-MTA	Embedded MTA
GRUU	Globally Routable User Agent URI
LNP	Local Number Portability
MSO	Member Service Operators
MTA	Media Terminal Adapter
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RST	Residential SIP Telephony
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
SBE	Signaling Path Border Element
SCF	Selective Call Forwarding
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SSP	SIP Service Provider
UA	User Agent
UAC	UA Client
UAS	User Agent Server
URI	Uniform Resource Identifier

5 OVERVIEW

Figure 1 shows the peering relationship between two SSP (SIP Service Provider) networks; Cable MSO-A and Cable MSO-B. The two peering MSO networks may be any combination of PacketCable 1.5 and PacketCable 2.0. The Signaling Path Border Element (SBE) serves as the egress/ingress point for SIP signaling into each peer network. The SBE may act as a proxy or a Back-to-Back User Agent (B2BUA). The optional Data Path Border Element (DBE) serves as a media relay at the peering interface for media interworking, topology hiding, and IPv4/6 interworking. When the DBE is not deployed, media are exchanged directly with the E-MTA (Embedded Media Terminal Adapter) or E-DVA (Embedded Digital Voice Adapter).

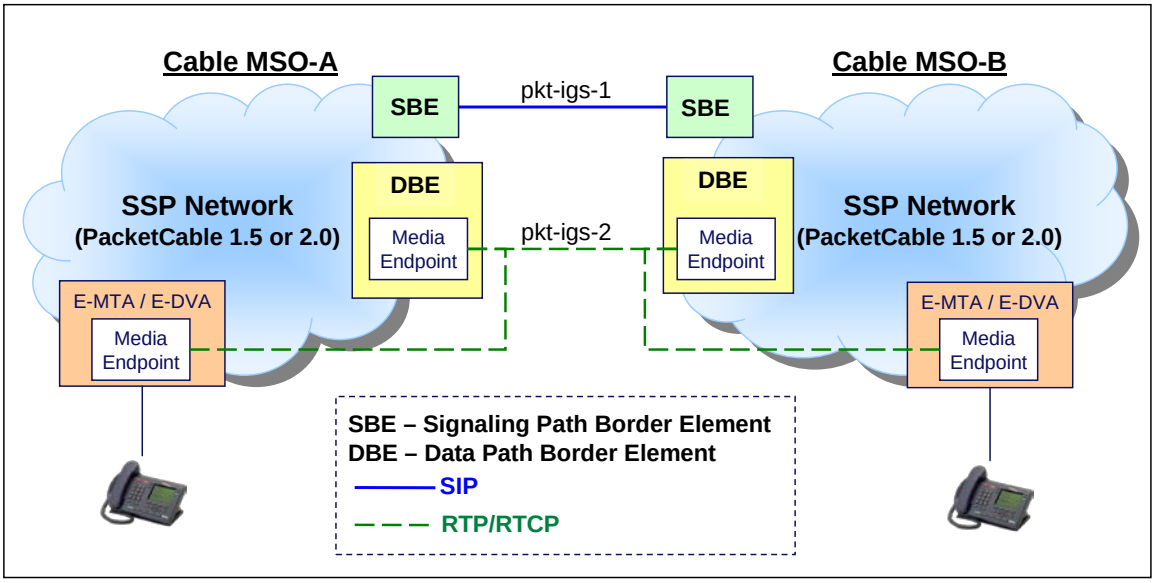

Figure 1 - Peering Architecture

As shown in Figure 1, this specification defines two reference points at the peering interface; pkt-igs-1 and pkt-igs-2. Pkt-igs-1 carries the SIP (Session Initiation Protocol) and SDP (Session Description Protocol) signaling between the peering networks, while pkt-igs-2 carries the media RTP (Real-Time Transport Protocol) and RTCP (RTP Control Protocol) packets between the peering networks.

Even though the pkt-igs-1 reference point terminates at the SBE of each peering SSP network, the responsibility for supporting this interface does not rest solely with the SBE. The reason for this is that the SBE can play the role of a firewall or a SIP Proxy that is largely transparent to the SIP signaling exchanged between the two networks. In reality, the responsibility for supporting the pkt-igs-1 belongs to the set of SIP entities in the SIP signaling chain, including the SBE plus the SIP proxies and UAs inside the SSP network. Therefore, normative statements in this document that define SIP and SDP requirements should be interpreted as applying to the set of SIP entities in the SIP signaling chain within the SSP network that can affect the peering interface pkt-igs-1. This document does not identify which specific entities within the SIP signaling chain are responsible for meeting the normative requirements.

As shown in Figure 1, pkt-igs-2 terminates at a Media Endpoint in the SSP network. However, the location of the Media Endpoint varies based on how the network is engineered. If a DBE is deployed and configured to perform media encode/decode, then pkt-igs-2 terminates at a Media Endpoint in the DBE. If the DBE is not deployed, or is configured to be transparent to RTP/RTCP (i.e., no encode/decode), then the pkt-igs-2 terminates at a Media Endpoint somewhere within the SP network; e.g., in the E-MTA or E-DVA (as shown in Figure 1) or in a Media Server.

6 GENERAL PROCEDURES

6.1 Extension Negotiation

SIP entities involved in session peering SHOULD be configured in such a way that they do not require any SIP extensions to be supported by the peer SSP (SIP Service Provider) network. When sending an out-of-dialog request to a peer SSP network, SIP entities involved in session peering SHOULD include a Supported header field identifying all the extensions supported by the sending network.

SIP entities involved in session peering MAY support configuration controls to disable certain extensions based on bilateral agreement between peer SSP networks. For example, a SIP entity involved in session peering could be configured to remove '100rel' from the Supported header in order to disable the use of reliable provisionable response (PRACK).

Note: Policies that limit or block the use of SIP extensions should be applied with care, since their application tends to disable SIP's native extension negotiation mechanism, and therefore inhibit the deployment of new services.

When sending a dialog-initiating request to a peer SSP network, SIP entities involved in session peering MUST identify all supported SIP requests in the Allow header field.

6.2 Public User Identities

Users are identified at the peering interface by their Public User Identity. A SIP entity involved in session peering MUST encode Public User Identities as a SIP URI of the telephone-subscriber syntax form of a Tel URI as indicated by the "user=phone" parameter (see [RFC 3261] section 19.1.6), where the user part of the SIP URI contains a global Tel URI as defined in [RFC 3966].

Example:

sip:+13035551212@example.operator.com;user=phone

6.2.1 Identifying the Called User

When sending a dialog-initiating request to a peer SSP network, SIP entities involved in session peering MUST:

- identify the called user in the Request-URI of the request, and
- identify the called user using the telephone-subscriber syntax form of the SIP URI as described above in Section 6.2.

In addition, if Local Number Portability (LNP) information for the called number was obtained, then SIP entities involved in session peering MUST:

- include the LNP data in SIP URI in the Request-URI using the Tel URI "npdi" and "rn" parameters as defined in [RFC 4694], and
- if the called number is ported, then identify the routing number using the global form of the "rn" parameter, which is indicated by a leading "+" character followed by the country-code followed by the national number (e.g., "rn=+16132220000").

On receiving a dialog-initiating request from a peer SSP network, SIP entities involved in session peering MUST:

- identify the called user based on the contents in the Request-URI, where the Request-URI contains a SIP URI as described above in Section 6.2;
- obtain the LNP data for the called number based on the presence and contents of the "npdi" and "rn" Tel URI parameters contained in the SIP URI in the Request-URI as defined in [RFC 4694].

Table 1 summarizes the allowed forms for the called Public User Identity at the peering interface.

Table 1 - Called Public User Identities

Use Case	Direction	Valid Form	Example
No LNP query	send/receive	SIP URI containing global Tel URI	sip:+13036614567@example.mso-a.com;user=phone
LNP Query - number not ported	send/receive	Above plus "npdi" parameter	sip:+13036614567;npdi@example.mso-a.com;user=phone
LNP Query - number ported	send/receive	Above plus global "rn" parameter	sip:+13036614567;npdi,rn=+13036620000@example.mso-a.com;user=phone

6.2.2 Identifying the Calling User

When sending or receiving a dialog-initiating request, SIP entities involved in session peering MUST identify the calling user in the P-Asserted-Identity header field using the telephone-subscriber syntax form of the SIP URI as described above in Section 6.2. When sending or receiving a dialog-initiating request, SIP entities involved in session peering SHOULD identify the calling name display information in the display-name component of the P-Asserted-Identity header field as described in Section 7.2.

6.3 Trust Domain and Asserted Identities

In a peering relationship, both originating and terminating SSP networks are in the same trust domain. Therefore, per [RFC 3325], the terminating SSP network MUST trust an originating peer SSP network to populate the P-Asserted-Identity header field in an incoming INVITE request with the Public User Identity of the originating user. Furthermore, the originating SSP network MUST trust the terminating SSP network to honor the privacy wishes of the originator, as indicated in the Privacy header field per [RFC 3323].

6.4 IPv4/6 Interworking

It is the responsibility of the IPv6 SSP network to perform the IPv4/IPv6 interworking function when interworking with an IPv4 SSP network.

6.5 Fault Isolation and Recovery

6.5.1 Interface Failure Detection

An SSP network MAY periodically send an OPTIONS request containing a Max-Forwards header field set to a value of '0' to detect the availability of a peer's ingress point. The ping rate is based on bi-lateral agreement (typically every 5 seconds). If the sending SSP network fails to receive a response to an OPTIONS request, then it will consider that non-responding ingress point into the peer SSP network to have failed, and will remove the ingress point from the available set of ingress points to the peer SSP network. When a failure is detected, the SSP network that detected the failure should attempt to route subsequent calls to the peer SSP network over an available alternate

route, with the final alternate route being the PSTN. In the meantime, the SSP network that detected the failure will continue to send periodic OPTIONS pings to the failed ingress point, in order to detect when it has been restored and is available for service.

Note: A possible enhancement to the OPTIONS ping is to declare a well-known SIP URI in the registry that could be used to test the health of each ingress point in a peer SSP network. For example, SIP INVITE (with no SDP) to SIP:999999999@mso-a.com would respond with a 200OK (again no SDP), followed by a BYE/200OK.

6.5.2 Congestion Control

SIP does not currently provide an explicit congestion control mechanism. However, an SSP network MAY impose limits on the number of simultaneous calls, and the incoming rate at which it will accept calls, from a peer. On receiving a dialog-initiating request that exceeds such limits, the receiving SSP network MUST respond with a 503 (Service Unavailable) response. An SSP network receiving a dialog-initiating request MUST limit the use of the 503 responses to reporting congestion at the ingress point. A terminating SSP network MUST NOT send a 503 response to an originating SSP network to report congestion or other failures that are internal to the terminating SSP network. For example, a 503 response generated by a SIP signaling entity within a terminating SSP network should be consumed within the terminating network, and should not be propagated across the peering interface to the originating SSP network (i.e., avoid sending a 503 response to an originating peer if the same failure is likely to be encountered when the call is retried via an alternate route).

On receiving a 503 (Service Unavailable) response from a peer SSP network, the receiving SSP network MUST limit the scope of the response to the call on which it was received (i.e., a 503 response has no affect on the routing of subsequent calls to the peer). Also, the receiving SSP network MUST attempt to consume the 503 response from a peer as close to the egress signaling point as possible, and avoid propagating the response back toward the originating CMS or E-DVA. Specifically, on receiving a 503 response to a dialog-initiating request that was sent to a peer SSP network, the receiving SSP network MUST:

- terminate the current transaction,
- ignore the Retry-After header field if one is present, and
- attempt to route the call via an alternate peering interface (i.e., do not attempt to route the call via the same peering interface since it may encounter and aggravate the same overload condition).

6.5.3 Session Timer

SIP entities involved in session peering SHOULD support Session Timer as defined in [RFC 4028].

6.5.4 RTP Loopback Test

Peer SSP networks SHOULD support the RTP Loopback Test procedures defined in [E-DVA]. SSP networks that support the RTP Loopback procedures will provide a SIP URI that identifies a media endpoint within the SSP network that performs the loopback functions. Ideally, this "loopback" media endpoint would be located near the ingress point of the peer SSP network.

6.6 Media

SIP entities involved in session peering MUST support the G.711 PCMU audio codec at a packetization interval of 20 msec as defined in [RFC 3551].

SIP entities involved in session peering MAY support voice-band-data relay mechanisms such as the following:

- T.38 fax relay as specified in [T.38]
- V.152 as specified in [V.152]
- DTMF-relay for digits 0-9 and * and # as defined in [RFC 4733]

A SIP entity involved in session peering that supports one or more of these voice-band-data relay mechanisms MUST revert to G.711 pass-though when interworking with a peer SSP network that does not support the same voice-band-date relay mechanism.

7 CALL FEATURES

7.1 Session Establishment

7.1.1 SDP Requirements

SIP entities involved in session peering MUST support the SDP requirements defined in [RFC 4566]. A SIP entity involved in session peering MUST include only one media (m=) descriptor per desired media stream in an SDP offer to a peer SSP network.

If a SIP entity involved in session peering receives an SDP offer containing multiple media descriptors, it MUST act on the media descriptors and include all of them in the same order in the response, including non-zero ports and zero ports for the offered media according to its capabilities as specified in [RFC 3264], an Offer/Answer Model with SDP. A SIP entity involved in session peering MUST NOT reject an offered session because it offers more media than the SIP entity can handle.

7.1.2 Basic Call Setup

This section describes the procedures at the peering interface required to establish a 2-way session for a basic voice call. In this case it is assumed that no originating or terminating features are applied (no call blocking, forwarding, etc), and that the called line is available to accept the call. Also, this section describes the session establishment procedures when the call is initiated by the originating SIP User Agent itself, and not via a 3rd party in support of features like click-to-call. Two-way call establishment using 3rd Party Call Control (3PCC) procedures is covered in Section 7.1.5.

SIP entities involved in session peering MUST support the SDP offer/answer procedures specified in [RFC 3264]. The originating SSP network MUST include an SDP offer in the initial INVITE. The terminating SSP network MUST include an SDP answer in the final 200 (OK) response to INVITE. The terminating SSP network MAY also include an SDP body in a provisional 18x response to INVITE. The SDP contained in an 18x provisional response can be considered a "preview" of the actual SDP answer to be sent in the 200 (OK) to INVITE. The originating SSP network can act on this "preview" SDP to establish an early media session, as described in Section 7.1.3. The terminating SSP network MUST ensure that the "preview" SDP matches the actual SDP answer contained in the 200 (OK) response to INVITE.

Note: An SDP offer/answer exchange occurs within the context of a single dialog. Therefore, the requirement for matching SDPs in the provisional and final responses to INVITE applies only when the provisional and final response are in the same dialog. If the provisional and final response are on different dialogs (say, when the INVITE is forked), the requirement for matching SDPs does not apply.

SIP entities involved in session peering MUST always set the SDP mode attribute in the initial offer/answer to "a=sendrecv".

Note: Setting the mode to "a=sendrecv" on the initial SDP offer/answer exchange avoids an additional SDP offer/answer exchange to update the mode to send-receive after the call is answered. This should help mitigate the problem of voice-clipping on answer.

SIP entities involved in session peering that advertise support for different but overlapping sets of codecs in the SDP offer/answer exchange for a given call MUST negotiate a common codec for the call.

7.1.3 Ringback Tone vs. Early Media

During the call setup phase, while the originating SSP network is waiting for the terminating SSP network to answer the call, the originating line is either playing local ringback tone to the calling user, or is connected to a receive-only

or bi-directional early-media session with the terminating SSP network. For example, early media can be supplied by the terminating endpoint (e.g., custom ringback tone) while waiting for answer.

SIP entities involved in session peering must use the following procedures to control whether the originating line applies local ringback tone or establishes an early media session while waiting for the call to be answered.

1. The terminating SSP network controls the application of local ringback tone at the originating line or the establishment of an early media session by sending the following provisional response to a call-initiating INVITE.
   - The terminating SSP Network MUST send a 180 (Alerting) response containing no SDP to the originating SP network, if the call scenario requires the application of local ringback tone at the originating line.
   - The terminating SSP Network MUST send a 183 (Progressing) response containing SDP that describes the terminating media endpoint to the originating SSP network, if the call scenario requires an early media session.
   - The provisional response sent for other call scenarios is not be specified, as long as the response is not one of those described above.
2. The originating SSP network performs the following action on receipt of a provisional response to a call-initiating INVITE.
   - The originating SSP network MUST apply local ringback tone if it receives a 180 (Alerting) response containing no SDP.
   - The originating SSP network MUST establish an early media session with the media endpoint described by the SDP when it receives a 18x response containing SDP.
   - The originating SSP Network MUST do nothing (e.g., continue to apply local ringback tone if it was already being applied when the response was received) if it receives a 18x response other than 180 (Alerting), and the response contains no SDP.

When establishing an early media session, the originating SSP network MAY immediately remove any local ringback tone currently being applied. Alternatively, the originating SSP network MAY wait for receipt of RTP that matches the received SDP, and apply other checks/policies to validate the received RTP, before removing any locally applied ringback tone.

7.1.4 Early-Media with Multiple Terminating Endpoints

There are some call scenarios that require media sessions to be established (serially) between the originating line and one or more intermediate media endpoints before the call is connected to the final target called party. For example, the terminating SSP network can insert a media server in the call to interact with the calling user in some way (e.g., to collect a blocking-override PIN) before offering the call to the called user. Another case occurs when the called user fails to answer within an allotted time and the call is redirected to voice-mail, or forwarded to another user via Call Forwarding Don't Answer (CFDA). These different cases can be combined in the same call.

For each terminating media endpoint that is associated with a call before the call is answered, the terminating SSP network must decide whether to establish an early media session, or apply ringback tone at the originating line. For example, consider the case where the called user has call blocking with PIN override, and CFDA. First, an early-media session is established with the call-blocking server to collect the PIN. Next, the originating line in instructed to play local ring-back tone while waiting for the called user to answer, and finally an early media session is established with the forward-to party to play custom ringback tone.

[RFC 3261] mandates that the SDP included in provisional 18x responses to INVITE within the context of a dialog must match the SDP-answer included in the final 200 (OK) response to INVITE. The following sections describe

two different mechanisms for supporting multiple terminating media endpoints before answer, within the confines of this requirement.

7.1.4.1 Forking the INVITE

For each terminating media endpoint that requires an early media session to be established with the originating line, the terminating SSP network MUST signal the attributes of the terminating media endpoint to the originating SSP network within the SDP of a 183 (Progressing) response. The terminating SSP network MUST ensure that 18x responses containing different SDP copies are not sent within the same dialog. The terminating SSP network does this by specifying a different tag parameter in the To header field for each provisional response that contains a unique SDP, as if the INVITE had been sequentially forked.

The originating SSP network MUST honor the most recently received 18x response to INVITE, based on the procedures defined in Section 7.1.3.

7.1.4.2 Redirecting the INVITE

As an alternative to sequentially forking the INVITE, the terminating entity can redirect the originating entity to the next endpoint in the series by sending a 302 (Moved Temporarily) response containing a Contact header field that identifies the next endpoint. The resulting INVITE from the originating SSP network is sent as a dialog-initiating request, and can therefore establish a new early-media session with the next endpoint in the series. The use of this procedure is based on bilateral agreement between peering operators.

On receiving a 302 (Moved Temporarily) response to an INVITE request, and if this mechanism is enabled based on local policy, the originating SSP network MUST send a new dialog-initiating INVITE with a Request-URI set to the value returned in the Contact header field of the 302 (Moved Temporarily) response, as described in [RFC 3261].

7.1.5 Establishing calls using 3PCC

Section 7.1.2 describes the procedures that are used to establish basic two-way call when the call is initiated directly by the originating user's endpoint. However, an SSP may support features such as click-to-call, where the call is initiated by a 3rd party such as an Application Server on behalf of the originating user. To support such features, SIP entities involved in session peering MUST support the 3PCC procedures described in [RFC 3725].

7.1.6 Hold

A SIP entity involved in session peering that wishes to place a media stream "on hold" MUST offer an updated SDP to its peer SSP network with an attribute of "a=inactive" or "a=sendonly" in the media description block. A SIP entity involved in session peering that wishes to place a media stream "on hold" MUST NOT set the connection information of the SDP to a null IP address. For example, the SIP entity involved in session peering MUST NOT set the 'c=' connection line to c=IN IP4 0.0.0.0. A SIP entity involved in session peering that wants to place a media stream "on hold" SHOULD locally mute the media stream.

A SIP entity involved in session peering that receives an SDP offer with an attribute of "a=inactive" in the media block MUST place the media stream "on hold" and send an SDP answer containing a media attribute of "a=inactive". A SIP entity involved in session peering that receives an SDP offer with an attribute of "a=inactive" in the media block MUST NOT set the connection data of the answer SDP to c=0.0.0.0. A SIP entity involved in session peering operating in IPv4 that receives an SDP offer with no directionality attributes but connection data set to c=IN IP4 0.0.0.0 SHOULD place the media stream "on hold".

7.2 Calling Name and Number Deliver (with Privacy)

The originating SSP network MUST provide the calling number of the originating user in the P-Asserted-Identity header field of dialog-initiating requests. Subject to local policies/agreements, the originating SSP network SHOULD provide the calling name of the originating user in the P-Asserted-Identity header field of dialog-initiating requests. (The mechanism for obtaining the calling name is outside the scope of this document.) The calling number is contained in the telephone-subscriber syntax form of the SIP URI, containing an E.164 number [E.164] as described in Section 6.2. The calling name is contained in the display-name component of the P-Asserted-Identity header field.

If the originating user wants to remain anonymous, the originating SSP network MUST include a Privacy header field containing the value "id" as specified in [RFC 3323] and [RFC 3325]. In addition, the originating SSP network SHOULD obscure the identity of the originating user in other header fields as follows:

- Set the identity information in the From header field to "Anonymous <sip:anonymous@anonymous.invalid>"
- Set the display-name in the To header field to "Anonymous" (since the To display-name selected by the originating user could provide a hint to the originating user's identity)
- Obscure any information from the Call-ID and Contact header fields, such as the originating FQDN, that could provide a hint to the originating user's identity

The terminating SSP network MUST obtain the calling name and number for caller-ID display from the contents of the P-Asserted-Identity header field contained in dialog-initiating requests. If the INVITE request contains a Privacy header with the value "id", the terminating SSP network MUST provide a display of "Private" to the terminating user.

7.3 Call Forwarding

If an SSP offers call-forwarding services to its users, then the forwarding SSP network MAY remain in the signaling path of the forwarded call in order to support separate billing for forward-from and forward-to legs. An SSP network that is required to remain in the signaling path of a forwarded call based on local policy MUST do so using one of the following procedures:

1. forward the INVITE to the forward-to-user while remaining in the signaling path as a SIP Proxy or B2BUA, or
2. respond to the initial INVITE with a 302 (Moved Temporarily) response with a Contact header field containing a private URI that points back to the forwarding SSP network.

7.3.1 Detecting Call-Forwarding Loops

A call forwarding loop is defined to be the scenario that occurs when a targeted subscriber for a call forwards the call to another destination. If the forwarded-to destination also has call forwarding configured, the call can forward back (directly or indirectly) to the original targeted subscriber. When a loop is detected, the entity that found the loop rejects the call. A SIP entity involved in session peering MUST detect call forwarding loops. A SIP entity involved in session peering MUST support a configurable limit on the number of times an individual call may be subject to forwarding. If the number of forwarding attempts for a single call exceeds this limit, the SIP entity involved in session peering MUST reject the call.

A SIP entity involved in session peering SHOULD detect call forwarding loops and limit the number times a call is forwarded by supporting the History-Info header field as defined in [RFC 4244], and by analyzing the History-Info header field entries as described in this section. However, these mechanisms alone may not be sufficient to detect loops when calls are forwarded to networks not supporting these mechanisms. Therefore, a SIP entity involved in

session peering MAY support additional loop prevention and forwarding limit detection methods as long as the requirements of forwarding limit restrictions and loop detection are met.

If the SIP entity involved in session peering supports the prevention of forwarding loops via analysis of the History-Info header field present in the INVITE request, then it MUST compare the forward-to address with the set of targeted-to URI (hi-targeted-to-uri) entries from the History-Info header field. If there is a match, then a loop has occurred. If no History-Info header field is present, then it is not possible to perform loop detection via this mechanism.

If a SIP entity involved in session peering supports the prevention of forwarding loops by enforcing a maximum number of forwarding attempts, then it MUST calculate the number of forwarding attempts by counting the number of entries in the History-Info header field that were added due to call forwarding (i.e., entries containing a nested Reason header which includes a protocol-cause parameter and a reason-text parameter that indicate the call was forwarded as defined below). If no History-Info header field is present, then it is not possible to determine the number of forwarding attempts via this mechanism.

[RFC 4458] defines the mapping between the forwarding conditions and the coding of the protocol-cause parameter in the Reason header. A SIP entity involved in session peering MUST populate the Reason header field with a protocol-cause value of "486" and a reason-text value of "CFBL" when the forwarding condition is Call Forwarding Busy Line (CFBL). The SIP entity involved in session peering MUST populate the Reason header field with a protocol-cause value of "408" and a reason-text value of "CFDA" when the forwarding condition is Call Forwarding Don't Answer (CFDA). The SIP entity involved in session peering MUST populate the Reason header field with a protocol-cause value of "302" and a reason-text value of "CFV/SCF" when the forwarding condition is Call Forwarding Variable (CFV) or Selective Call Forwarding (SCF).

7.4 Call Transfer

A user in a peered call can perform the various forms of call-transfer (e.g., consultative transfer, blind transfer). Call-transfer can be supported in one of two ways; either using the REFER request and Replaces header, or by manipulating the call legs using 3rd Party Call Control (3PCC) techniques. SIP entities involved in session peering that support call transfer MUST support the 3PCC option, and MAY support the REFER/Replaces option. If an SSP network supports both options, then the option that is used when interworking with a specific peer is based on locally configured data that indicates the capabilities of that peer.

7.4.1 Call Transfer using REFER/Replaces

SIP entities involved in session peering that support call-transfer using the procedures described in the section MUST support the SIP REFER extension described in [RFC 3515], and the SIP Replaces extension described in [RFC 3891]. Furthermore, [RFC 3515] requires support of the SIP Event Notification extension described in [RFC 3265].

To describe the basic transfer call-flow, consider the case where user-A in SSP network-A is in an active call with user-B in peered SSP network-B, and user-A decides to transfer user-B to user-C. User-C could be located anywhere in the global network; for example in network-A, network-B, another peered SSP network, a non-peering IP network, or the PSTN. Here are the basic steps to complete the transfer using REFER/Replaces:

1. User-A puts user-B on hold (sends re-INVITE with SDP "a=inactive" as described in Section 7.1.6).
2. User-A initiates a basic 2-way call to user-C.
3. User-A sends an in-dialog REFER to user-B containing a Refer-To header field. The Refer-To header field instructs user-B to send an INVITE to user-C with an imbedded Replaces header field identifying the A-to-C dialog.

- If SSP network-A is not required to remain in the signaling path of the transferred call, then it identifies user-C directly in the Refer-To header field,
- If SSP network-A is required to remain in the signaling path of the transferred call (say to generate events for proper billing of the call), then it identifies a private URL pointing to itself in the Refer-To header field, as described in [RFC 3603].

4. User-B sends an INVITE containing the Replaces header field specified in step 3 to the address contained in the Refer-To header field (i.e., the INVITE is routed to user-C either directly from SSP network-B, or indirectly via SSP network-A using the private URL).
5. User-B sends NOTIFY requests within the original A-to-B dialog, informing user-A of the progress of the B-to-C call.
6. At some point user-A drops out of both dialogs (e.g., drops out of A-to-C dialog on receiving BYE from user-C). At this point users B and C are active in a 2-way call.

SIP entities involved in session peering SHOULD support receiving a Globally Routable User Agent URI (GRUU) as defined in [RFC 5627] in the Refer-To header.

7.4.2 Call Transfer Using 3PCC

SIP entities involved in session peering that support call-transfer using 3PCC techniques MUST act as a B2BUA, and manipulate the call legs using INVITE and re-INVITE requests. It is recommended that such techniques follow the guidance presented in [RFC 3725].

7.5 3-Way Conference

The media mixing for 3-way conference calls may be performed by the E-MTA or E-DVA endpoint of the conference control party, or by a conference bridge server in the peer SSP network serving the conference control party. When mixing is done by the E-MTA or E-DVA endpoint, there are no specific requirements placed on the peering interface other than the support of media hold as described in Section 7.1.6. When conference mixing is performed by a network-based server, users are added to the conference using procedures similar to those described for call transfer in Section 7.4.

7.6 Auto Recall/Callback

When a user invokes Auto-Callback, (AC) or Auto-Recall, (AR) and the user targeted by the recall/callback feature belongs to a peer SSP network, the originating SSP network first attempts to establish a basic 2-way call with the target user. If the call completes normally (e.g., the target user answers) then the feature is complete. If the terminating SSP network responds with an indication that the target user is busy, then the originating SSP network subscribes to the dialog-event package as defined in [RFC 4235] of the target user, as a mechanism to detect when the target user becomes available. When the terminating SSP network subsequently notifies the originating SSP network that the target user is available, the originating SSP network re-attempts to establish a 2-way call to the target user.

7.6.1 Originating SSP Network Sends INVITE to Target

When a user invokes an AR or AC call, the originating SSP network MUST follow the procedures given for a basic call as described in Section 7.1.2, and attempt to establish a 2-way call with the target user. In addition, the originating SSP network MUST add a Call-Info header field to the INVITE with a purpose of "answer_if_not_busy".

If the originating SSP network receives a 200-OK response to INVITE, then the AC/AR feature is considered complete, and the remainder of the call is handled like a normal 2-way call. If the originating SSP network receives a

486-Busy-Here or 600-Busy-Everywhere response to the INVITE, then it MUST follow the AC/AR procedures as defined below. If the terminating SSP network receives an inbound INVITE with a Call-Info header field declaring purpose=answer_if_not_busy, then the terminating SSP network MUST ignore any active Call-Forwarding-Busy-Line (CFBL) service for the target user, not forward the call if the target is busy, and instead handle the call as if CFBL was not active (e.g., offer the call using the call-waiting feature).

7.6.2 Originating SSP Network Sends SUBSCRIBE to Target

On receiving a 486-Busy-Here or 600-Busy-Everywhere response to an AC/AR INVITE request, the originating SSP network MUST establish a subscription to the dialog event package of the target endpoint, by sending a SUBSCRIBE request containing an Event header field set to "dialog" to the terminating SSP network. The originating SSP network MUST populate the SUBSCRIBE Request-URI with the URI returned in the Contact header field of the INVITE response, if that URI is a Globally Routable User Agent URI (GRUU), as defined in [RFC 5627]. Otherwise, the originating SSP network MUST populate the Request-URI with the Public User Identity of the target callback/recall user.

7.6.3 Target Sends NOTIFY to Originating SSP Network

On receiving the SUBSCRIBE to the dialog event package, the terminating SSP network MUST notify the originating SSP network of the dialog state of the target user endpoint as described in [RFC 4235]. Upon receiving a NOTIFY message of "target is idle", the originating SSP network MUST first cancel the dialog-event subscription by sending a SUBSCRIBE message with an Expires header field containing the value "0".

Once the subscription is cancelled, the originating SSP network MUST send a new INVITE request to establish a call with the target user. If the originating SSP network receives a 486-Busy-Here or 600-Busy-Everywhere response to the INVITE, then it MUST automatically re-subscribe to the dialog event package of the target user.

Note: A "busy" response could be returned in this case as a result of a race condition, where the target endpoint sends a NOTIFY of "target is idle", and then becomes busy in a new call before the subsequent INVITE is received).

Annex A Response Codes

This annex documents the semantics for the common response codes that appear on the peering interface so an SSP network that receives a response code from a peer will take the correct action.

Table 2 lists response codes for some of the common call failures. For many of the 4xx error cases, the response code would only be generated for the stated condition if the call wasn't handled in some manner by the terminating SSP network (e.g., call routed to voice mail).

Table 2 - Response Codes

Condition	Response Code	Example Action when Received
Endpoint is unavailable • MTA powered down • MTA removed from service by OS • Line in lockout	480 Temporarily Unavailable	Reorder tone, or announcement "Your call cannot be completed at this time. Please hang up and try again later."
Line is "busy" • Line doesn't have call waiting and is busy in a call • Line has call waiting, but is already busy with two calls, busy in an emergency call, is in a transient state with another call (ringing, origination glare, etc)	486 Busy Here	Busy tone
Call times out waiting for user action • Ringing timeout waiting for answer • Timeout waiting to accept call-waiting call • Timeout waiting for caller to enter digits after solicitor-call-blocking prompt	480 Temporarily Unavailable	Reorder tone, or announcement "Your call cannot be completed at this time. Please hang up and try again later."
Call blocked by a feature • Terminating call blocking • Do not disturb	403 Forbidden	Announcement: "Due to network difficulties, your call cannot be completed at this time. Please try your call again later."
Call blocked because called user not authorized to receive calls • Temporarily disconnected due to late payment • Recently deleted	404 Not Found	Announcement: "Your call cannot be completed as dialed. Please check the number and try again."
Call blocked due to resource limitation • No QoS • MTA resource exhaustion (e.g., no DSP resources)	480 Temporarily Unavailable	Reorder tone, or announcement "Your call cannot be completed at this time. Please hang up and try again later."
Call Forward loop detected	Depends on type of call forwarding: • CFBL: 486 Busy Here • CFDA, CFV, SCF: 480 Temporary Failure	Reorder tone, or announcement "Your call cannot be completed at this time. Please hang up and try again later."

Condition	Response Code	Example Action when Received
During call-transfer, transfer-to user agent can't find dialog identified in Replaces header	481 Call/Transaction Doesn't Exist	Application dependent
Called endpoint can not support SDP offer • Does not support IP version in SDP c= line • Does not support any offered codec • Not authorized for authored media	488 Not Acceptable Here	Reorder, or announcement
Called address does not exist • Target routing number not owned by this network • Called user does not exist in this network	404 Not Found	Announcement: "Your call cannot be completed as dialed. Please check the number and try again."
Congestion encountered at the peering interface	503 Service Unavailable	Retry call via PSTN (see Section 6.5.2 for more details).

Appendix I Acknowledgements

CableLabs wishes to thank David Hancock for the development of this specification.

PacketCable Specifications Team