           Connectivity Preconditions for Session Description Protocol (SDP)
                              Media Streams

Abstract

   This document defines a new connectivity precondition for the Session
   Description Protocol (SDP) precondition framework.  A connectivity
   precondition can be used to delay session establishment or
   modification until media stream connectivity has been successfully
   verified.  The method of verification may vary depending on the type
   of transport used for the media.  For unreliable datagram transports
   such as UDP, verification involves probing the stream with data or
   control packets.  For reliable connection-oriented transports such as
   TCP, verification can be achieved simply by successful connection
   establishment or by probing the connection with data or control
   packets, depending on the situation.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF).  It represents the consensus of the IETF community.  It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG).  Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc5898.

Copyright Notice

Table of Contents

1.  Introduction

   The concept of a Session Description Protocol (SDP) [RFC4566]
   precondition in the Session Initiation Protocol (SIP) [RFC3261] is
   defined in RFC 3312 [RFC3312] (updated by RFC 4032 [RFC4032]).  A
   precondition is a condition that has to be satisfied for a given
   media stream in order for session establishment or modification to
   proceed.  When the precondition is not met, session progress is
   delayed until the precondition is satisfied or the session
   establishment fails.  For example, RFC 3312 [RFC3312] defines the
   Quality of Service precondition, which is used to ensure availability
   of network resources prior to establishing a session (i.e., prior to
   starting to alert the callee).

   SIP sessions are typically established in order to set up one or more
   media streams.  Even though a media stream may be negotiated
   successfully through an SDP offer-answer exchange, the actual media
   stream itself may fail.  For example, when there is one or more
   Network Address Translators (NATs) or firewalls in the media path,
   the media stream may not be received by the far end.  In cases where
   the media is carried over a connection-oriented transport such as TCP
   [RFC0793], the connection-establishment procedures may fail.  The
   connectivity precondition defined in this document ensures that
   session progress is delayed until media stream connectivity has been
   verified.

The connectivity precondition type defined in this document follows
the guidelines provided in RFC 4032 [RFC4032] to extend the SIP
preconditions framework.

## 2.  Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [RFC2119].

## 3.  Connectivity Precondition Definition

## 3.1.  Syntax

The connectivity precondition type is defined by the string "conn",
and hence we modify the grammar found in RFC 3312 [RFC3312] and RFC
5027 [RFC5027] as follows:

    precondition-type = "conn" / "sec" / "qos" / token

This precondition tag is registered with the IANA in Section 8.

## 3.2.  Operational Semantics

According to RFC 4032 [RFC4032], documents defining new precondition
types need to describe the behavior of UAs (User Agents) from the
moment session establishment is suspended due to a set of
preconditions, until it is resumed when these preconditions are met.
An entity that wishes to delay session establishment or modification
until media stream connectivity has been established uses this
precondition-type in an offer.  When a mandatory connectivity
precondition is received in an offer, session establishment or
modification is delayed until the connectivity precondition has been
met (i.e., until media stream connectivity has been established in
the desired direction or directions).  The delay of session
establishment defined here implies that alerting of the called party
does not occur until the precondition has been satisfied.

Packets may be both sent and received on the media streams in
question.  However, such packets SHOULD be limited to packets that
are necessary to verify connectivity between the two endpoints
involved on the media stream.  That is, the underlying media stream
SHOULD NOT be cut through.  For example, Interactive Connectivity
Establishment (ICE) connectivity checks [RFC5245] and TCP SYN, SYN-
ACK, and ACK packets can be exchanged on media streams that support
them as a way of verifying connectivity.

Some media streams are described by a single 'm' line but,
nevertheless, involve multiple addresses.  For example, RFC 5109
[RFC5109] specifies how to send FEC (Forward Error Correction)
information as a separate stream (the address for the FEC stream is
provided in an 'a=fmtp' line).  When a media stream consists of
multiple destination addresses, connectivity to all of them MUST be
verified in order for the precondition to be met.  In the case of RTP
media streams [RFC3550] that use RTCP, connectivity MUST be verified
for both RTP and RTCP; the RTCP transmission interval rules MUST
still be adhered to.

## 3.3.  Status Type

RFC 3312 [RFC3312] defines support for two kinds of status types --
namely, segmented and end-to-end.  The connectivity precondition-type
defined here MUST be used with the end-to-end status type; use of the
segmented status type is undefined.

## 3.4.  Direction Tag

The direction attributes defined in RFC 3312 [RFC3312] are
interpreted as follows:

o   send: the party that generated the session description is sending
    packets on the media stream to the other party, and the other
    party has received at least one of those packets.  That is, there
    is connectivity in the forward (sending) direction.

o   recv: the other party is sending packets on the media stream to
    the party that generated the session description, and this party
    has received at least one of those packets.  That is, there is
    connectivity in the backwards (receiving) direction.

o   sendrecv: both the send and recv conditions hold.

Note that a "send" connectivity precondition from the offerer's point
of view corresponds to a "recv" connectivity precondition from the
answerer's point of view, and vice versa.  If media stream
connectivity in both directions is required before session
establishment or modification continues, the desired status needs to
be set to "sendrecv".

## 3.5.  Precondition Strength

Connectivity preconditions may have a strength-tag of either
"mandatory" or "optional".

When a mandatory connectivity precondition is offered and the
answerer cannot satisfy the connectivity precondition (e.g., because
the offer does not include parameters that enable connectivity to be
verified without media cut through) the offer MUST be rejected as
described in RFC 3312 [RFC3312].

When an optional connectivity precondition is offered, the answerer
MUST generate its answer SDP as soon as possible.  Since session
progress is not delayed in this case, it is not known whether the
associated media streams will have connectivity.  If the answerer
wants to delay session progress until connectivity has been verified,
the answerer MUST increase the strength of the connectivity
precondition by using a strength-tag of "mandatory" in the answer.

Note that use of a mandatory precondition requires the presence of a
SIP "Require" header with the option tag "precondition".  Any SIP UA
that does not support a mandatory precondition will reject such
requests.  To get around this issue, an optional connectivity
precondition and the SIP "Supported" header with the option tag
"precondition" can be used instead.

Offers with connectivity preconditions in re-INVITEs or UPDATEs
follow the rules given in Section 6 of RFC 3312 [RFC3312].  That is:

    Both user agents SHOULD continue using the old session parameters
    until all the mandatory preconditions are met.  At that moment,
    the user agents can begin using the new session parameters.

4.  Verifying Connectivity

Media stream connectivity is ascertained by use of a connectivity
verification mechanism between the media endpoints.  A connectivity
verification mechanism may be an explicit mechanism, such as ICE
[RFC5245] or ICE TCP [ICE-TCP], or it may be an implicit mechanism,
such as TCP.  Explicit mechanisms provide specifications for when
connectivity between two endpoints using an offer/answer exchange is
ascertained, whereas implicit mechanisms do not.  The verification
mechanism is negotiated as part of the normal offer/answer exchange;
however, it is not identified explicitly.  More than one mechanism
may be negotiated, but the offerer and answerer need not use the
same.  The following rules guide which connectivity verification
mechanism to use:

o  If an explicit connectivity verification mechanism (e.g., ICE) is
   negotiated, the precondition is met when the mechanism verifies
   connectivity successfully.

   o  Otherwise, if a connection-oriented transport (e.g., TCP) is
      negotiated, the precondition is met when the connection is
      established.

   o  Otherwise, if an implicit verification mechanism is provided by
      the transport itself or the media stream data using the transport,
      the precondition is met when the mechanism verifies connectivity
      successfully.

   o  Otherwise, connectivity cannot be verified reliably, and the
      connectivity precondition will never be satisfied if requested.

   This document does not mandate any particular connectivity
   verification mechanism; however, in the following, we provide
   additional considerations for verification mechanisms.

4.1.  Correlation of Dialog to Media Stream

   SIP and SDP do not provide any inherent capabilities for associating
   an incoming media stream packet with a particular dialog.  Thus, when
   an offerer is trying to ascertain connectivity, and an incoming media
   stream packet is received, the offerer may not know which dialog had
   its "recv" connectivity verified.  Explicit connectivity verification
   mechanisms therefore typically provide a means to correlate the media
   stream, whose connectivity is being verified, with a particular SIP
   dialog.  However, some connectivity verification mechanisms may not
   provide such a correlation.  In the absence of a mechanism for the
   correlation of dialog to media stream (e.g., ICE), a UAS (User Agent
   Server) MUST NOT require the offerer to confirm a connectivity
   precondition.

4.2.  Explicit Connectivity Verification Mechanisms

   Explicit connectivity verification mechanisms typically use probe
   traffic with some sort of feedback to inform the sender whether
   reception was successful.  Below we provide two examples of such
   mechanisms, and how they are used with connectivity preconditions:

   Interactive Connectivity Establishment (ICE) [RFC5245] provides one
   or more candidate addresses in signaling between the offerer and the
   answerer and then uses STUN (Simple Traversal of the UDP Protocol
   through NAT) Binding Requests to determine which pairs of candidate
   addresses have connectivity.  Each STUN Binding Request contains a
   password that is communicated in the SDP as well; this enables
   correlation between STUN Binding Requests and candidate addresses for
   a particular media stream.  It also provides correlation with a
   particular SIP dialog.

ICE implementations may be either full or lite (see [RFC5245]).  Full
implementations generate and respond to STUN Binding Requests,
whereas lite implementations only respond to them.  With ICE, one
side is a controlling agent, and the other side is a controlled
agent.  A full implementation can take on either role, whereas a lite
implementation can only be a controlled agent.  The controlling agent
decides which valid candidate to use and informs the controlled agent
of it by identifying the pair as the nominated pair.  This leads to
the following connectivity precondition rules:

o  A full implementation ascertains both "send" and "recv"
   connectivity when it operates as a STUN client and has sent a STUN
   Binding Request that resulted in a successful check for all the
   components of the media stream (as defined further in ICE).

o  A full or a lite implementation ascertains "recv" connectivity
   when it operates as a STUN server and has received a STUN Binding
   Request that resulted in a successful response for all the
   components of the media stream (as defined further in ICE).

o  A lite implementation ascertains "send" and "recv" connectivity
   when the controlling agent has informed it of the nominated pair
   for all the components of the media stream.

A simpler and slightly more delay-prone alternative to the above
rules is for all ICE implementations to ascertain "send" and "recv"
connectivity for a media stream when the ICE state for that media
stream has moved to Completed.

Note that there is never a need for the answerer to request
confirmation of the connectivity precondition when using ICE: the
answerer can determine the status locally.  Also note, that when ICE
is used to verify connectivity preconditions, the precondition is not
satisfied until connectivity has been verified for all the component
transport addresses used by the media stream.  For example, with an
RTP-based media stream where RTCP is not suppressed, connectivity
MUST be ascertained for both RTP and RTCP.  Finally, it should be
noted, that although connectivity has been ascertained, a new offer/
answer exchange may be required before media can flow (per ICE).

The above are merely examples of explicit connectivity verification
mechanisms.  Other techniques can be used as well.  It is however
RECOMMENDED that ICE be supported by entities that support
connectivity preconditions.  Use of ICE has the benefit of working
for all media streams (not just RTP) as well as facilitating NAT and
firewall traversal, which may otherwise interfere with connectivity.
Furthermore, the ICE recommendation provides a baseline to ensure

that all entities that require probe traffic to support the
connectivity preconditions have a common way of ascertaining
connectivity.

## 4.3.  Verifying Connectivity for Connection-Oriented Transports

Connection-oriented transport protocols generally provide an implicit
connectivity verification mechanism.  Connection establishment
involves sending traffic in both directions thereby verifying
connectivity at the transport-protocol level.  When a three-way (or
more) handshake for connection establishment succeeds, bi-directional
communication is confirmed and both the "send" and "recv"
preconditions are satisfied whether requested or not.  In the case of
TCP for example, once the TCP three-way handshake has completed (SYN,
SYN-ACK, ACK), the TCP connection is established and data can be sent
and received by either party (i.e., both a send and a receive
connectivity precondition has been satisfied).  SCTP (Stream Control
Transmission Protocol) [RFC4960] connections have similar semantics
as TCP and SHOULD be treated the same.

When a connection-oriented transport is part of an offer, it may be
passive, active, or active/passive [RFC4145].  When it is passive,
the offerer expects the answerer to initiate the connection
establishment, and when it is active, the offerer wants to initiate
the connection establishment.  When it is active/passive, the
answerer decides.  As noted earlier, lack of a media-stream-to-dialog
correlation mechanism can make it difficult to guarantee with whom
connectivity has been ascertained.  When the offerer takes on the
passive role, the offerer will not necessarily know which SIP dialog
originated an incoming connection request.  If the offerer instead is
active, this problem is avoided.

## 5.  Connectivity and Other Precondition Types

The role of a connectivity precondition is to ascertain media stream
connectivity before establishing or modifying a session.  The
underlying intent is for the two parties to be able to exchange media
packets successfully.  However, connectivity by itself may not fully
satisfy this.  Quality of Service, for example, may be required for
the media stream; this can be addressed by use of the "qos"
precondition defined in RFC 3312 [RFC3312].  Similarly, successful
security parameter negotiation may be another prerequisite; this can
be addressed by use of the "sec" precondition defined in RFC 5027
[RFC5027].

## 6.  Examples

   The first example uses the connectivity precondition with TCP in the
   context of a session involving a wireless access medium.  Both UAs
   use a radio access network that does not allow them to send any data
   (not even a TCP SYN) until a radio bearer has been set up for the
   connection.  Figure 1 shows the message flow of this example (the
   required PRACK transaction has been omitted for clarity -- see
   [RFC3312] for details):

```
                    A                                          B
                    |   INVITE                                 |
                    |   a=curr:conn e2e none                   |
                    |   a=des:conn mandatory e2e sendrecv      |
                    |   a=setup:holdconn                       |
                    |----------------------------------------->|
                    |                                          |
                    |   183 Session Progress                   |
                    |   a=curr:conn e2e none                   |
                    |   a=des:conn mandatory e2e sendrecv      |
                    |   a=setup:holdconn                       |
                    |<-----------------------------------------|
                    |                                          |
                    |   UPDATE                                 |
                    |   a=curr:conn e2e none                   |
                    |   a=des:conn mandatory e2e sendrecv      |
        A's radio   |   a=setup:actpass                        |
        bearer is   +----------------------------------------->|
        up          |                                          |
                    |   200 OK                                 |
                    |   a=curr:conn e2e none                   |
                    |   a=des:conn mandatory e2e sendrecv      |
                    |   a=setup:active                         |
                    |<-----------------------------------------|
                    |                                          |
                    |                                          |
                    |                                          | B's radio
                    |<---TCP Connection Establishment--->+       bearer is up
                    |                                          | B sends TCP SYN
                    |                                          |
                    |                                          |
                    |   180 Ringing                            | TCP connection
                    |<-----------------------------------------+ is up
                    |                                          | B alerts the user
                    |                                          |
```
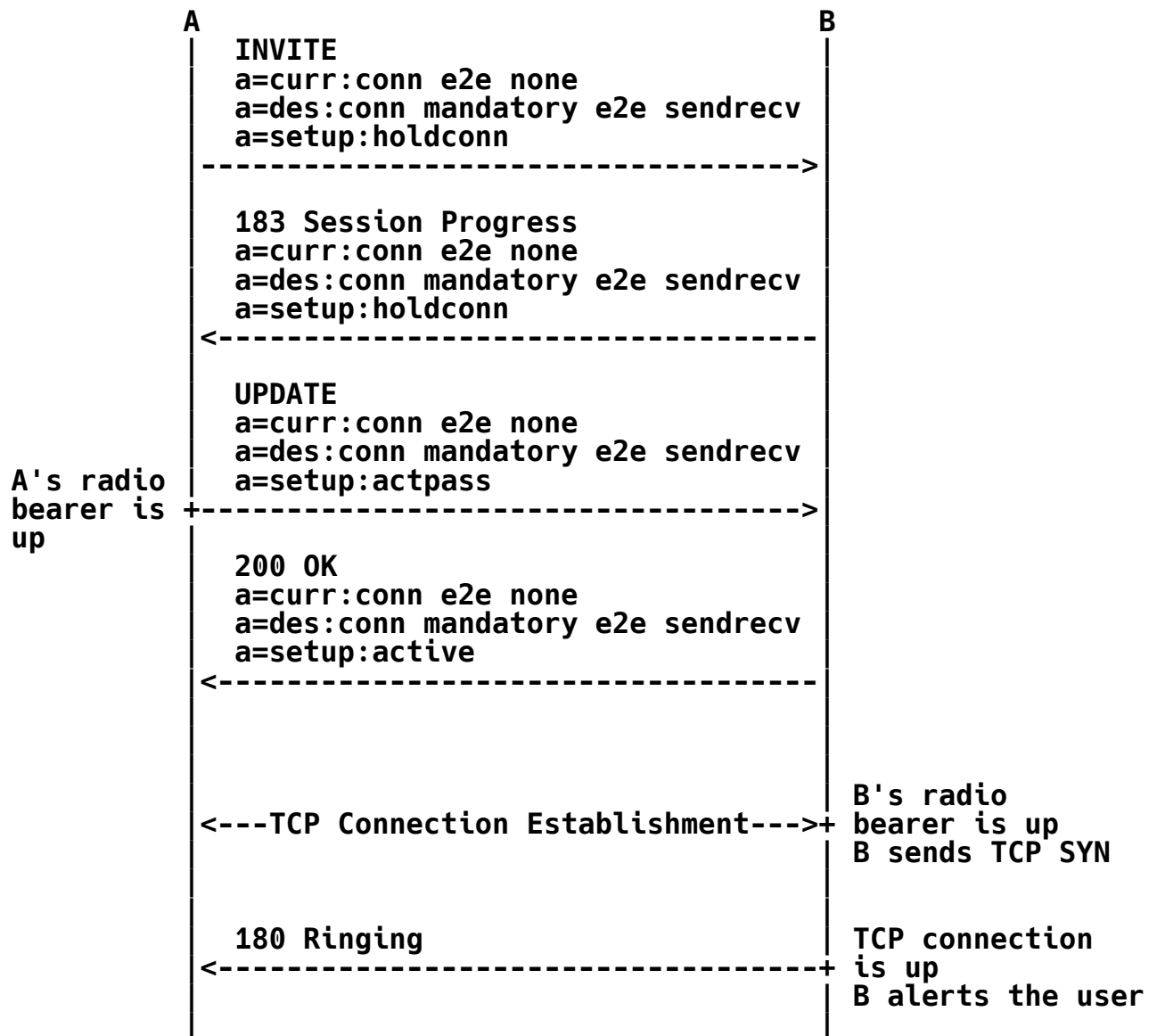
             Figure 1: Message Flow with Two Types of Preconditions

A sends an INVITE requesting connection-establishment preconditions.
The setup attribute in the offer is set to holdconn [RFC4145] because
A cannot send or receive any data before setting up a radio bearer
for the connection.

B agrees to use the connectivity precondition by sending a 183
(Session Progress) response.  The setup attribute in the answer is
also set to holdconn because B, like A, cannot send or receive any
data before setting up a radio bearer for the connection.

When A's radio bearer is ready, A sends an UPDATE to B with a setup
attribute with a value of actpass.  This attribute indicates that A
can perform an active or a passive TCP open.  A is letting B choose
which endpoint will initiate the connection.

Since B's radio bearer is not ready yet, B chooses to be the one
initiating the connection and indicates this with a setup attribute
with a value of active.  At a later point, when B's radio bearer is
ready, B initiates the TCP connection towards A.

Once the TCP connection is established successfully, B knows the
"sendrecv" precondition is satisfied, and B proceeds with the session
(i.e., alerts the Callee), and sends a 180 (Ringing) response.

The second example shows a basic SIP session establishment using SDP
connectivity preconditions and ICE (the required PRACK transaction
and some SDP details have been omitted for clarity).  The offerer (A)
is a full ICE implementation whereas the answerer (B) is a lite ICE
implementation.  The message flow for this scenario is shown in
Figure 2 below.

```
                     A                                              B
                     |                                              |
                     |-------------(1) INVITE SDP1---------------->|
                     |                                              |
                     |<------(2) 183 Session Progress SDP2--------|
                     |                                              |
                     |~~~~~ Connectivity check to B ~~~~~~~~~~~~~~>|
                     |<~~~~ Connectivity to B OK ~~~~~~~~~~~~~~~~~|
                     |                                              |
                     |-------------(3) UPDATE SDP3---------------->|
                     |                                              |
                     |<--------(4) 200 OK (UPDATE) SDP4-----------|
                     |                                              |
                     |<-------------(5) 180 Ringing--------------|
                     |                                              |
```
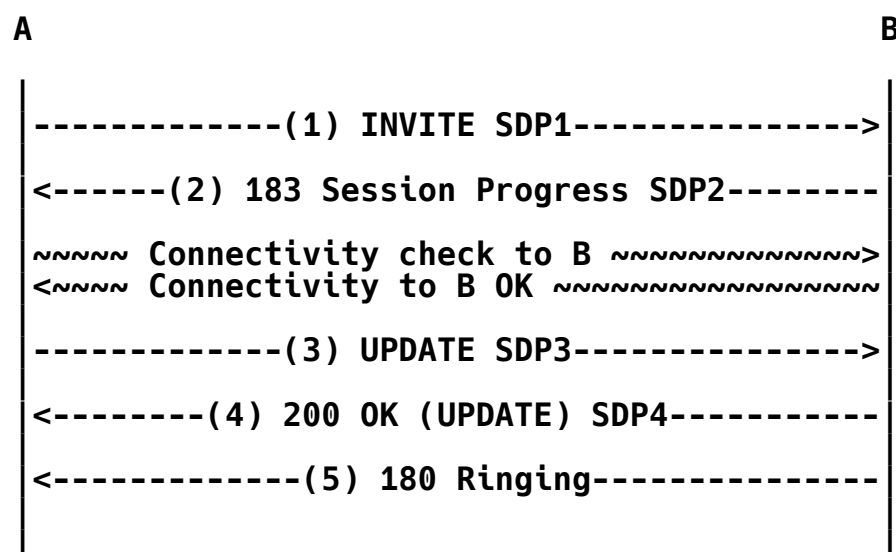
       Figure 2: Connectivity Precondition with ICE Connectivity Checks

   SDP1: A includes a mandatory end-to-end connectivity precondition
   with a desired status of "sendrecv"; this will ensure media stream
   connectivity in both directions before continuing with the session
   setup.  Since media stream connectivity in either direction is
   unknown at this point, the current status is set to "none".  A's
   local status table (see [RFC3312]) for the connectivity precondition
   is as follows:

| Direction | Current | Desired Strength | Confirm |
|-----------|---------|------------------|---------|
| send      | no      | mandatory        | no      |
| recv      | no      | mandatory        | no      |

   and the resulting offer SDP is:

```
     a=ice-pwd:asd88fgpdd777uzjYhagZg
     a=ice-ufrag:8hhY
     m=audio 20000 RTP/AVP 0
     c=IN IP4 192.0.2.1
     a=rtcp:20001
     a=curr:conn e2e none
     a=des:conn mandatory e2e sendrecv
     a=candidate:1 1 UDP 2130706431 192.0.2.1 20000 typ host
```

   SDP2: When B receives the offer, B sees the mandatory sendrecv
   connectivity precondition.  B is a lite ICE implementation and hence
   B can only ascertain "recv" connectivity (from B's point of view)

from A; thus, B wants A to inform it about connectivity in the other
direction ("send" from B's point of view).  B's local status table
therefore looks as follows:

| Direction | Current | Desired Strength | Confirm |
|-----------|---------|------------------|---------|
| send | no | mandatory | no |
| recv | no | mandatory | no |

Since B is a lite ICE implementation and B wants to ask A for
confirmation about the "send" (from B's point of view) connectivity
precondition, the resulting answer SDP becomes:

```
a=ice-lite
a=ice-pwd:qrCA8800133321zF9AIj98
a=ice-ufrag:H92p
m=audio 30000 RTP/AVP 0
c=IN IP4 192.0.2.4
a=rtcp:30001
a=curr:conn e2e none
a=des:conn mandatory e2e sendrecv
a=conf:conn e2e send
a=candidate:1 1 UDP 2130706431 192.0.2.4 30000 typ host
```

Since the "send" and the "recv" connectivity precondition (from B's
point of view) are still not satisfied, session establishment remains
suspended.

SDP3: When A receives the answer SDP, A notes that B is a lite ICE
implementation and that confirmation was requested for B's "send"
connectivity precondition, which is the "recv" precondition from A's
point of view.  A performs a successful send and recv connectivity
check to B by sending an ICE connectivity check to B and receiving
the corresponding response.  A's local status table becomes:

| Direction | Current | Desired Strength | Confirm |
|-----------|---------|------------------|---------|
| send | yes | mandatory | no |
| recv | yes | mandatory | yes |

whereas B's local status table becomes:

| Direction | Current | Desired Strength | Confirm |
|-----------|---------|------------------|---------|
| send | no | mandatory | no |
| recv | yes | mandatory | no |

Since B asked for confirmation about the "recv" connectivity (from
A's point of view), A now sends an UPDATE (5) to B to confirm the
connectivity from A to B:

```
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 20000 RTP/AVP 0
c=IN IP4 192.0.2.1
a=rtcp:20001
a=curr:conn e2e sendrecv
a=des:conn mandatory e2e sendrecv
a=candidate:1 1 UDP 2130706431 192.0.2.1 20000 typ host
```

B knows it has recv connectivity (verified by ICE as well as A's
UPDATE) and send connectivity (confirmed by A's UPDATE) at this
point.  B's local status table becomes:

| Direction | Current | Desired Strength | Confirm |
|-----------|---------|------------------|---------|
| send | yes | mandatory | no |
| recv | yes | mandatory | no |

and the session can continue.

## 7.  Security Considerations

General security considerations for preconditions are discussed in
RFC 3312 [RFC3312] and RFC 4032 [RFC4032].  As discussed in RFC 4032
[RFC4032], it is strongly RECOMMENDED that S/MIME [RFC3853] integrity
protection be applied to the SDP session descriptions.  When the user
agent provides identity services (rather than the proxy server), the
SIP identity mechanism specified in RFC 4474 [RFC4474] provides an
alternative end-to-end integrity protection.  Additionally, the
following security issues relate specifically to connectivity
preconditions.

Connectivity preconditions rely on mechanisms beyond SDP, such as TCP
[RFC0793] connection establishment or ICE connectivity checks
[RFC5245], to establish and verify connectivity between an offerer
and an answerer.  An attacker that prevents those mechanisms from
succeeding (e.g., by keeping ICE connectivity checks from arriving at
their destination) can prevent media sessions from being established.
While this attack relates to connectivity preconditions, it is
actually an attack against the connection-establishment mechanisms
used by the endpoints.  This attack can be performed in the presence
or in the absence of connectivity preconditions.  In their presence,
the whole session setup will be disrupted.  In their absence, only
the establishment of the particular stream under attack will be

disrupted.  This specification does not provide any mechanism against
attackers able to block traffic between the endpoints involved in the
session because such an attacker will always be able to launch DoS
(Denial-of-Service) attacks.

Instead of blocking the connectivity checks, the attacker can
generate forged connectivity checks that would cause the endpoints to
assume that there was connectivity when there was actually no
connectivity.  This attack would result in the user experience being
poor because the session would be established without all the media
streams being ready.  The same attack can be used, regardless of
whether or not connectivity preconditions are used, to attempt to
hijack a connection.  The forged connectivity checks would trick the
endpoints into sending media to the wrong direction.  To prevent
these attacks, it is RECOMMENDED that the mechanisms used to check
connectivity are adequately secured by message authentication and
integrity protection.  For example, Section 2.5 of [RFC5245]
discusses how message integrity and data origin authentication are
implemented in ICE connectivity checks.

## 8.  IANA Considerations

IANA has registered a new precondition type under the Precondition
Types used with SIP subregistry, which is located under the Session
Initiation Protocol (SIP) Parameters registry.

| Precondition-Type | Description | Reference |
|-------------------|-------------------------------------|-----------|
| conn | Connectivity precondition | [RFC5898] |

## 9.  References

### 9.1.  Normative References

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261]   Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
            A., Peterson, J., Sparks, R., Handley, M., and E.
            Schooler, "SIP: Session Initiation Protocol", RFC 3261,
            June 2002.

[RFC3312]   Camarillo, G., Marshall, W., and J. Rosenberg,
            "Integration of Resource Management and Session Initiation
            Protocol (SIP)", RFC 3312, October 2002.

   [RFC3853]  Peterson, J., "S/MIME Advanced Encryption Standard (AES)
              Requirement for the Session Initiation Protocol (SIP)",
              RFC 3853, July 2004.

   [RFC4032]  Camarillo, G. and P. Kyzivat, "Update to the Session
              Initiation Protocol (SIP) Preconditions Framework",
              RFC 4032, March 2005.

   [RFC4474]  Peterson, J. and C. Jennings, "Enhancements for
              Authenticated Identity Management in the Session
              Initiation Protocol (SIP)", RFC 4474, August 2006.

   [RFC4566]  Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
              Description Protocol", RFC 4566, July 2006.

   [RFC5027]  Andreasen, F. and D. Wing, "Security Preconditions for
              Session Description Protocol (SDP) Media Streams",
              RFC 5027, October 2007.

   [RFC5245]  Rosenberg, J., "Interactive Connectivity Establishment
              (ICE): A Protocol for Network Address Translator (NAT)
              Traversal for Offer/Answer Protocols", RFC 5245,
              April 2010.

9.2.  Informative References

   [RFC0793]  Postel, J., "Transmission Control Protocol", STD 7,
              RFC 793, September 1981.

   [RFC3550]  Schulzrinne, H., Casner, S., Frederick, R., and V.
              Jacobson, "RTP: A Transport Protocol for Real-Time
              Applications", STD 64, RFC 3550, July 2003.

   [RFC4145]  Yon, D. and G. Camarillo, "TCP-Based Media Transport in
              the Session Description Protocol (SDP)", RFC 4145,
              September 2005.

   [RFC4960]  Stewart, R., "Stream Control Transmission Protocol",
              RFC 4960, September 2007.

   [RFC5109]  Li, A., "RTP Payload Format for Generic Forward Error
              Correction", RFC 5109, December 2007.

   [ICE-TCP]  Perreault, S., Ed. and J. Rosenberg, "TCP Candidates with
              Interactive Connectivity Establishment (ICE)", Work
              in Progress, October 2009.

Authors' Addresses

   Flemming Andreasen
   Cisco Systems, Inc.
   499 Thornall Street, 8th Floor
   Edison, NJ  08837
   USA

   EMail: fandreas@cisco.com


   Gonzalo Camarillo
   Ericsson
   Hirsalantie 11
   Jorvas  02420
   Finland

   EMail: Gonzalo.Camarillo@ericsson.com


   David Oran
   Cisco Systems, Inc.
   7 Ladyslipper Lane
   Acton, MA  01720
   USA

   EMail: oran@cisco.com


   Dan Wing
   Cisco Systems, Inc.
   170 West Tasman Drive
   San Jose, CA  95134
   USA

   EMail: dwing@cisco.com