                  NAT Traversal Practices for Client-Server SIP

Abstract

   Traversal of the Session Initiation Protocol (SIP) and the sessions
   it establishes through Network Address Translators (NATs) is a
   complex problem.  Currently, there are many deployment scenarios and
   traversal mechanisms for media traffic.  This document provides
   concrete recommendations and a unified method for NAT traversal as
   well as documents corresponding flows.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   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).  Not all documents
   approved by the IESG are a candidate for any level of Internet
   Standard; see 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/rfc6314.

   to this document.  Code Components extracted from this document must
   include Simplified BSD License text as described in Section 4.e of
   the Trust Legal Provisions and are provided without warranty as
   described in the Simplified BSD License.

Table of Contents

1.  Introduction

   NAT (Network Address Translator) traversal has long been identified
   as a complex problem when considered in the context of the Session
   Initiation Protocol (SIP) [RFC3261] and its associated media such as
   the Real-time Transport Protocol (RTP) [RFC3550].  The problem is
   exacerbated by the variety of NATs that are available in the
   marketplace today and the large number of potential deployment
   scenarios.  Details of different NATs behavior can be found in "NAT
   Behavioral Requirements for Unicast UDP" [RFC4787].

   The IETF has been active on many specifications for the traversal of
   NATs, including Session Traversal Utilities for NAT (STUN) [RFC5389],
   Interactive Connectivity Establishment (ICE) [RFC5245], symmetric
   response [RFC3581], symmetric RTP [RFC4961], Traversal Using Relay
   NAT (TURN) [RFC5766], SIP Outbound [RFC5626], the Session Description
   Protocol (SDP) attribute for RTP Control Protocol (RTCP) [RFC3605],
   "Multiplexing RTP Data and Control Packets on a Single Port"
   [RFC5761], and others.  Each of these represents a part of the
   solution, but none of them gives the overall context for how the NAT
   traversal problem is decomposed and solved through this collection of
   specifications.  This document serves to meet that need.  It should
   be noted that this document intentionally does not invoke 'Best
   Current Practice' machinery as defined in RFC 2026 [RFC2026].

   The document is split into two distinct sections as follows:

   o  Section 4 provides a definitive set of best common practices to
      demonstrate the traversal of SIP and its associated media through
      NAT devices.

   o  Section 5 provides non-normative examples representing
      interactions of SIP using various NAT type deployments.

   The document does not propose any new functionality but does draw on
   existing solutions for both core SIP signaling and media traversal
   (as defined in Section 4).

   The best practices described in this document are for traditional
   "client-server"-style SIP.  This term refers to the traditional use
   of the SIP protocol where User Agents talk to a series of
   intermediaries on a path to connect to a remote User Agent.  It seems
   likely that other groups using SIP, for example, peer-to-peer SIP
   (P2PSIP), will recommend these same practices between a P2PSIP client
   and a P2PSIP peer, but will recommend different practices for use
   between peers in a peer-to-peer network.

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].

   It should be noted that the use of the term 'Endpoint-Independent
   NAT' in this document refers to a NAT that is both Endpoint-
   Independent Filtering and Endpoint-Independent Mapping per the
   definitions in RFC 4787 [RFC4787].

3.  Problem Statement

   The traversal of SIP through NATs can be split into two categories
   that both require attention: the core SIP signaling and associated
   media traversal.  This document assumes NATs that do not contain SIP-
   aware Application Layer Gateways (ALGs), which makes much of the
   issues discussed in the document not applicable.  ALGs have
   limitations (as per RFC 4787 [RFC4787] Section 7, RFC 3424 [RFC3424],
   and [RFC5245] Section 18.6), and experience shows they can have an
   adverse impact on the functionality of SIP.  This includes problems
   such as requiring the media and signaling to traverse the same device
   and not working with encrypted signaling and/or payload.

   The use of non-TURN-based media intermediaries is not considered in
   this document.  More information can be obtained from [RFC5853] and
   [MIDDLEBOXES].

   The core SIP signaling has a number of issues when traversing through
   NATs.

   SIP response routing over UDP as defined in RFC 3261 [RFC3261]
   without extensions causes the response to be delivered to the source
   IP address specified in the topmost Via header, or the 'received'
   parameter of the topmost 'Via' header.  The port is extracted from
   the SIP 'Via' header to complete the IP address/port combination for
   returning the SIP response.  While the destination for the response
   is correct, the port contained in the SIP 'Via' header represents the
   listening port of the originating client and not the port
   representing the open pinhole on the NAT.  This results in responses
   being sent back to the NAT but to a port that is likely not open for
   SIP traffic.  The SIP response will then be dropped at the NAT.  This
   is illustrated in Figure 1, which depicts a SIP response being
   returned to port 5060.

```
      Private                       NAT                    Public
      Network                        |                     Network
                                     |
                                     |
      --------         SIP Request   |open port 10923        --------
     |        |--------------------->--->----------------------|        |
     |        |                      |                       |        |
     | Client |                      |port 5060   SIP Response | Proxy  |
     |        |                      x<----------------------|        |
     |        |                      |                       |        |
      --------                       |                        --------
                                     |
                                     |
                                     |
                                     |
```
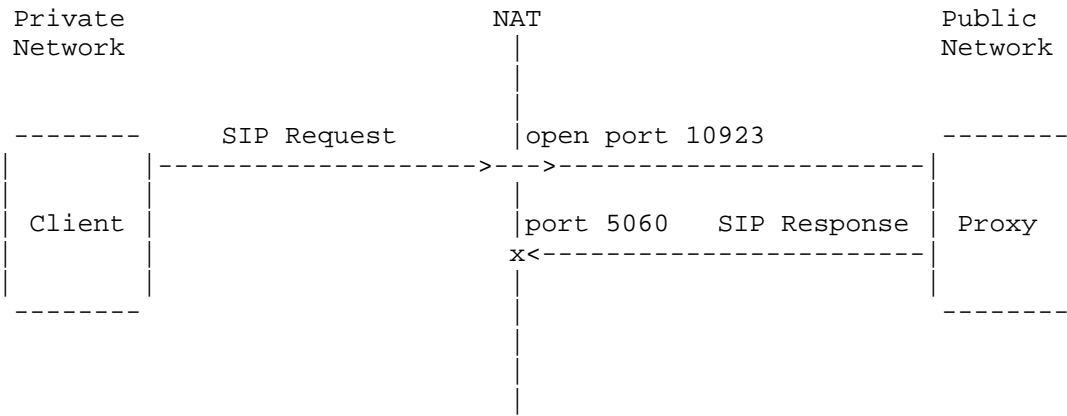
                     Figure 1: Failed Response

   Secondly, there are two cases where new requests reuse existing
   connections.  The first is when using a reliable, connection-oriented
   transport protocol such as TCP, SIP has an inherent mechanism that
   results in SIP responses reusing the connection that was created/used
   for the corresponding transactional request.  The SIP protocol does
   not provide a mechanism that allows new requests generated in the
   reverse direction of the originating client to use, for example, the
   existing TCP connection created between the client and the server
   during registration.  This results in the registered contact address
   not being bound to the "connection" in the case of TCP.  Requests are
   then blocked at the NAT, as illustrated in Figure 2.  The second case
   is when using an unreliable transport protocol such as UDP where
   external NAT mappings need to be reused to reach a SIP entity on the
   private side of the network.

```
      Private                       NAT                    Public
      Network                        |                     Network
                                     |
                                     |
      -------- (UAC 8023)   REGISTER/Response      (UAS 5060) --------
     |        |--------------------->---<----------------------|        |
     |        |                      |                       |        |
     | Client |                      |5060  INVITE   (UAC 8015)| Proxy  |
     |        |                      x<----------------------|        |
     |        |                      |                       |        |
      --------                       |                        --------
                                     |
                                     |
                                     |
                                     |
```
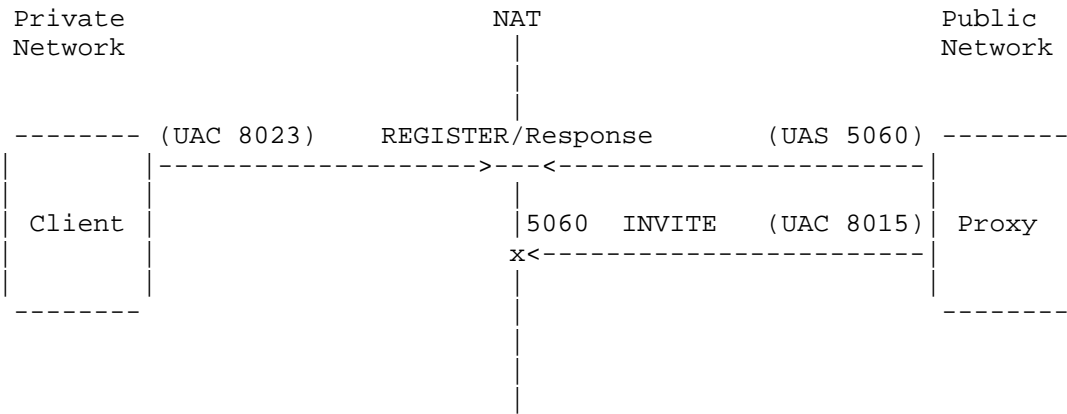
                     Figure 2: Failed Request

In Figure 2, the original REGISTER request is sent from the client on
port 8023 and received by the proxy on port 5060, establishing a
connection and opening a pinhole in the NAT.  The generation of a new
request from the proxy results in a request destined for the
registered entity (contact IP address) that is not reachable from the
public network.  This results in the new SIP request attempting to
create a connection to a private network address.  This problem would
be solved if the original connection were reused.  While this problem
has been discussed in the context of connection-oriented protocols
such as TCP, the problem exists for SIP signaling using any transport
protocol.  The impact of connection reuse of connection-oriented
transports (TCP, TLS, etc.) is discussed in more detail in the
connection reuse specification [RFC5923].  The approach proposed for
this problem in Section 4 of this document is relevant for all SIP
signaling in conjunction with connection reuse, regardless of the
transport protocol.

NAT policy can dictate that connections should be closed after a
period of inactivity.  This period of inactivity may vary from a
number of seconds to hours.  SIP signaling cannot be relied upon to
keep connections alive for the following two reasons.  Firstly, SIP
entities can sometimes have no signaling traffic for long periods of
time, which has the potential to exceed the inactivity timer, and
this can lead to problems where endpoints are not available to
receive incoming requests as the connection has been closed.
Secondly, if a low inactivity timer is specified, SIP signaling is
not appropriate as a keep-alive mechanism as it has the potential to
add a large amount of traffic to the network, which uses up valuable
resources and also requires processing at a SIP stack, which is also
a waste of processing resources.

Media associated with SIP calls also has problems traversing NAT.
RTP [RFC3550] runs over UDP and is one of the most common media
transport types used in SIP signaling.  Negotiation of RTP occurs
with a SIP session establishment using the Session Description
Protocol (SDP) [RFC4566] and a SIP offer/answer exchange [RFC3264].
During a SIP offer/answer exchange, an IP address and port
combination are specified by each client in a session as a means of
receiving media such as RTP.  The problem arises when a client
advertises its address to receive media and it exists in a private
network that is not accessible from outside the NAT.  Figure 3
illustrates this problem.

```
             NAT                 Public Network           NAT
              |                                            |
              |                                            |
              |                                            |
     --------  |             SIP Signaling Session    |  --------
    |        | |---------------------->Proxy<------------------| |        |
    |        | |  |                                       | |  |        |
    | Client | |  |                                       | |  | Client |
    |   A    | |>======>RTP>==Unknown Address==>X         | |  |   B    |
    |        | |  |              X<==Unknown Address==<RTP<===<| |        |
     --------  |  |                                         |  --------
              |                                            |
              |                                            |
              |                                            |
              |                                            |
```
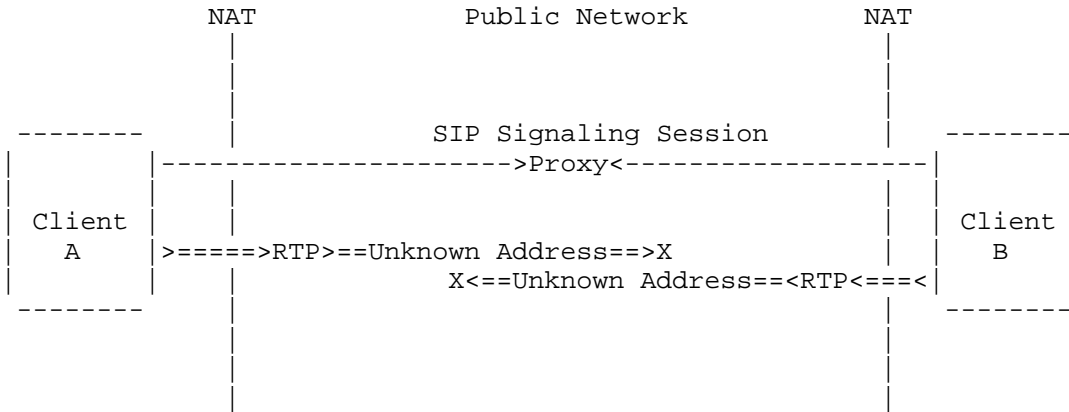
                        Figure 3: Failed Media

The connection addresses of the clients behind the NATs will
nominally contain a private IPv4 address that is not routable across
the public Internet.  Exacerbating matters even more would be the
tendency of Client A to send media to a destination address it
received in the signaling confirmation message -- an address that may
actually correspond to a host within the private network who is
suddenly faced with incoming RTP packets (likewise, Client B may send
media to a host within its private network who did not solicit these
packets).  Finally, to complicate the problem even further, a number
of different NAT topologies with different default behaviors
increases the difficulty of arriving at a unified approach.  This
problem exists for all media transport protocols that might be NATted
(e.g., TCP, UDP, the Stream Control Transmission Protocol (SCTP), the
Datagram Congestion Control Protocol (DCCP)).

In general, the problems associated with NAT traversal can be
categorized as follows.

For signaling:

o  Responses do not reuse the NAT mapping and filtering entries
   created by the request.

o  Inbound requests are filtered out by the NAT because there is no
   long-term connection between the client and the proxy.

   For media:

   o  Each endpoint has a variety of addresses that can be used to reach
      it (e.g., native interface address, public NATted address).  In
      different situations, a different pair of (local endpoint, remote
      endpoint) addresses should be used, and it is not clear when to
      use which pair.

   o  Many NATs filter inbound packets if the local endpoint has not
      recently sent an outbound packet to the sender.

   o  Classic RTCP usage is to run RTCP on the next highest port.
      However, NATs do not necessarily preserve port adjacency.

   o  Classic RTP and RTCP usage is to use different 5-tuples for
      traffic in each direction.  Though not really a problem, doing
      this through NATs is more work than using the same 5-tuple in both
      directions.

4.  Solution Technology Outline Description

   As mentioned previously, the traversal of SIP through existing NATs
   can be divided into two discrete problem areas: getting the SIP
   signaling across NATs and enabling media as specified by SDP in a SIP
   offer/answer exchange to flow between endpoints.

4.1.  SIP Signaling

   SIP signaling has two areas that result in transactional failure when
   traversing through NATs, as described in Section 3 of this document.
   The remaining sub-sections describe appropriate solutions that result
   in SIP signaling traversal through NATs, regardless of transport
   protocol.  It is advised that SIP-compliant entities follow the
   guidelines presented in this section to enable traversal of SIP
   signaling through NATs.

4.1.1.  Symmetric Response

   As described in Section 3 of this document, when using an unreliable
   transport protocol such as UDP, SIP responses are sent to the IP
   address and port combination contained in the SIP 'Via' header field
   (or default port for the appropriate transport protocol if not
   present).  Figure 4 illustrates the response traversal through the
   open pinhole using Symmetric techniques defined in RFC 3581
   [RFC3581].

```
        Private                      NAT                    Public
        Network                       |                     Network
                                      |
                                      |
        --------                      |                     --------
       |        |                     |                    |        |
       |        |send request------------------------------------->|        |
       | Client |<------------------------------send response|  SIP  |
       |   A    |                     |                    | Proxy  |
       |        |                     |                    |        |
        --------                      |                     --------
                                      |
                                      |
                                      |
```
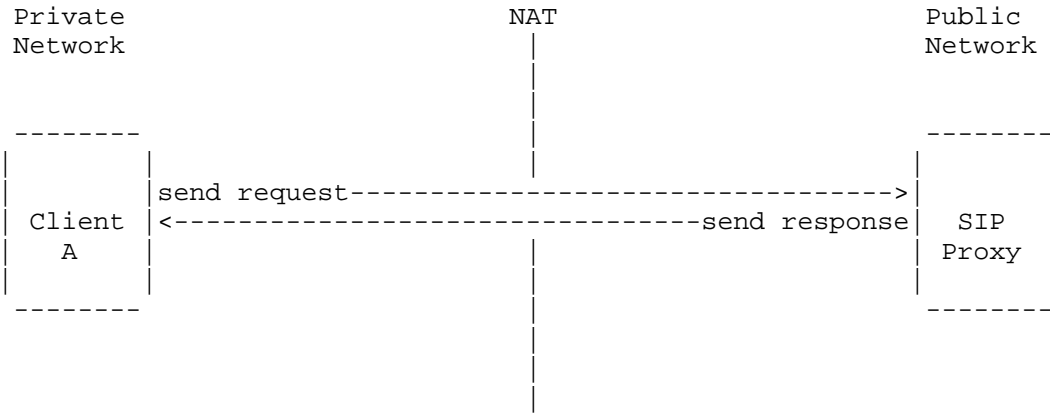
                    Figure 4: Symmetric Response

   The outgoing request from Client A opens a pinhole in the NAT.  The
   SIP Proxy would normally respond to the port available in the SIP
   'Via' header, as illustrated in Figure 1.  The SIP Proxy honors the
   'rport' parameter in the SIP 'Via' header and routes the response to
   the port from which it was sent.  The exact functionality for this
   method of response traversal is called 'Symmetric Response', and the
   details are documented in RFC 3581 [RFC3581].  Additional
   requirements are imposed on SIP entities in RFC 3581 [RFC3581] such
   as listening and sending SIP requests/responses from the same port.

4.1.2.  Client-Initiated Connections

   The second problem with SIP signaling, as defined in Section 3 and
   illustrated in Figure 2, is to allow incoming requests to be properly
   routed.

   Guidelines for devices such as User Agents that can only generate
   outbound connections through NATs are documented in "Managing Client-
   Initiated Connections in the Session Initiation Protocol (SIP)"
   [RFC5626].  The document provides techniques that use a unique User
   Agent instance identifier (instance-id) in association with a flow
   identifier (reg-id).  The combination of the two identifiers provides
   a key to a particular connection (both UDP and TCP) that is stored in
   association with registration bindings.  On receiving an incoming
   request to a SIP Address-Of-Record (AOR), a proxy/registrar routes to
   the associated flow created by the registration and thus a route
   through NATs.  It also provides a keep-alive mechanism for clients to
   keep NAT bindings alive.  This is achieved by multiplexing a ping-
   pong mechanism over the SIP signaling connection (STUN for UDP and

CRLF/operating system keepalive for reliable transports like TCP).
Usage of [RFC5626] is RECOMMENDED.  This mechanism is not transport
specific and should be used for any transport protocol.

Even if the SIP Outbound mechanism is not used, clients generating
SIP requests SHOULD use the same IP address and port (i.e., socket)
for both transmission and receipt of SIP messages.  Doing so allows
for the vast majority of industry provided solutions to properly
function (e.g., NAT traversal that is Session Border Control (SBC)
hosted).  Deployments should also consider the mechanism described in
the Connection Reuse [RFC5923] specification for routing
bidirectional messages securely between trusted SIP Proxy servers.

## 4.2.  Media Traversal

The issues of media traversal through NATs is not straightforward and
requires the combination of a number of traversal methodologies.  The
technologies outlined in the remainder of this section provide the
required solution set.

### 4.2.1.  Symmetric RTP/RTCP

The primary problem identified in Section 3 of this document is that
internal IP address/port combinations cannot be reached from the
public side of NATs.  In the case of media such as RTP, this will
result in no audio traversing NATs (as illustrated in Figure 3).  To
overcome this problem, a technique called 'Symmetric RTP/RTCP'
[RFC4961] can be used.  This involves a SIP endpoint both sending and
receiving RTP/RTCP traffic from the same IP address/port combination.
When operating behind a NAT and using the 'latching' technique
described in [MIDDLEBOXES], SIP User Agents MUST implement Symmetric
RTP/RTCP.  This allows traversal of RTP across the NAT.

### 4.2.2.  RTCP

Normal practice when selecting a port for defining RTP Control
Protocol (RTCP) [RFC3550] is for consecutive-order numbering (i.e.,
select an incremented port for RTCP from that used for RTP).  This
assumption causes RTCP traffic to break when traversing certain types
of NATs due to various reasons (e.g., already allocated port,
randomized port allocation).  To combat this problem, a specific
address and port need to be specified in the SDP rather than relying
on such assumptions.  RFC 3605 [RFC3605] defines an SDP attribute
that is included to explicitly specify transport connection
information for RTCP so a separate, explicit NAT binding can be set
up for the purpose.  The address details can be obtained using any
appropriate method including those detailed in this section (e.g.,
STUN, TURN, ICE).

A further enhancement to RFC 3605 [RFC3605] is defined in [RFC5761],
which specifies 'muxing' both RTP and RTCP on the same IP/PORT
combination.

4.2.3.  STUN/TURN/ICE

   ICE, STUN, and TURN are a suite of 3 inter-related protocols that
   combine to provide a complete media traversal solution for NATs.  The
   following sections provide details of each component part.

4.2.3.1.  STUN

   Session Traversal Utilities for NAT or STUN is defined in RFC 5389
   [RFC5389].  STUN is a lightweight tool kit and protocol that provides
   details of the external IP address/port combination used by the NAT
   device to represent the internal entity on the public facing side of
   NATs.  On learning of such an external representation, a client can
   use it accordingly as the connection address in SDP to provide NAT
   traversal.  Using terminology defined in "NAT Behavioral Requirements
   for Unicast UDP" [RFC4787], STUN does work with Endpoint-Independent
   Mapping but does not work with either Address-Dependent Mapping or
   Address and Port-Dependent Mapping type NATs.  Using STUN with either
   of the previous two NAT mappings to probe for the external IP
   address/port representation will provide a different result to that
   required for traversal by an alternative SIP entity.  The IP address/
   port combination deduced for the STUN server would be blocked for RTP
   packets from the remote SIP User Agent.

   As mentioned in Section 4.1.2, STUN is also used as a client-to-
   server keep-alive mechanism to refresh NAT bindings.

4.2.3.2.  TURN

   As described in Section 4.2.3.1, the STUN protocol does not work for
   UDP traversal through certain identified NAT mappings.  'Traversal
   Using Relays around NAT' is a usage of the STUN protocol for deriving
   (from a TURN server) an address that will be used to relay packets
   towards a client.  TURN provides an external address (globally
   routable) at a TURN server that will act as a media relay that
   attempts to allow traffic to reach the associated internal address.
   The full details of the TURN specification are defined in [RFC5766].
   A TURN service will almost always provide media traffic to a SIP
   entity, but it is RECOMMENDED that this method would only be used as
   a last resort and not as a general mechanism for NAT traversal.  This
   is because using TURN has high performance costs when relaying media
   traffic and can lead to unwanted latency.

4.2.3.3.  ICE

   Interactive Connectivity Establishment (ICE) is the RECOMMENDED
   method for traversal of existing NATs if Symmetric RTP and media
   latching are not sufficient.  ICE is a methodology for using existing
   technologies such as STUN, TURN, and any other protocol compliant
   with Unilateral Self-Address Fixing (NSAF) [RFC3424] to provide a
   unified solution.  This is achieved by obtaining as many
   representative IP address/port combinations as possible using
   technologies such as STUN/TURN (note: an ICE endpoint can also use
   other mechanisms (e.g., the NAT Port Mapping Protocol [NAT-PMP],
   Universal Plug and Play Internet Gateway Device [UPnP-IGD]) to learn
   public IP addresses and ports, and populate a=candidate lines with
   that information).  Once the addresses are accumulated, they are all
   included in the SDP exchange in a new media attribute called
   'candidate'.  Each candidate SDP attribute entry has detailed
   connection information including a media address, priority, and
   transport protocol.  The appropriate IP address/port combinations are
   used in the order specified by the priority.  A client compliant to
   the ICE specification will then locally run STUN servers on all
   addresses being advertised using ICE.  Each instance will undertake
   connectivity checks to ensure that a client can successfully receive
   media on the advertised address.  Only connections that pass the
   relevant connectivity checks are used for media exchange.  The full
   details of the ICE methodology are in [RFC5245].

5.  NAT Traversal Scenarios

   This section of the document includes detailed NAT traversal
   scenarios for both SIP signaling and the associated media.  Signaling
   NAT traversal is achieved using [RFC5626].

5.1.  Basic NAT SIP Signaling Traversal

   The following sub-sections concentrate on SIP signaling traversal of
   NATs.  The scenarios include traversal for both reliable and
   unreliable transport protocols.

5.1.1.  Registration (Registrar/Edge Proxy Co-Located)

   The set of scenarios in this section document basic signaling
   traversal of a SIP REGISTER method through NATs.

5.1.1.1.  UDP

```
                                            Registrar/
             Bob               NAT          Edge Proxy
              |                 |                |
              |(1) REGISTER     |                |
              |---------------->|                |
              |                 |                |
              |                 |(1) REGISTER    |
              |                 |--------------->|
              |                 |                |
              |*********************************|
              |   Create Outbound Connection Tuple   |
              |*********************************|
              |                 |                |
              |                 |(2) 200 OK      |
              |                 |<---------------|
              |                 |                |
              |(2) 200 OK       |                |
              |<----------------|                |
              |                 |                |
```
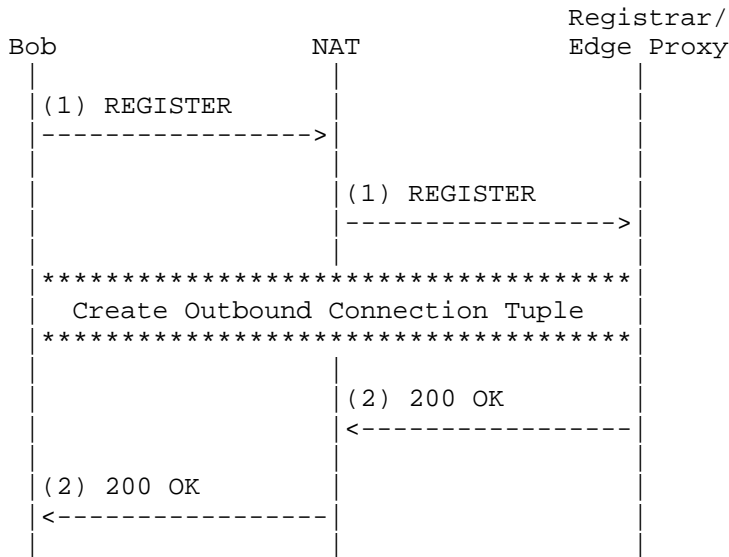
                    Figure 5: UDP Registration

   In this example, the client sends a SIP REGISTER request through a
   NAT.  The client will include an 'rport' parameter as described in
   Section 4.1.1 of this document for allowing traversal of UDP
   responses.  The original request as illustrated in (1) in Figure 5 is
   a standard SIP REGISTER message:

   Message 1:

   REGISTER sip:example.com SIP/2.0
   Via: SIP/2.0/UDP 192.168.1.2;rport;branch=z9hG4bKnashds7
   Max-Forwards: 70
   From: Bob <sip:bob@example.com>;tag=7F94778B653B
   To: Bob <sip:bob@example.com>
   Call-ID: 16CB75F21C70
   CSeq: 1 REGISTER
   Supported: path, outbound
   Contact: <sip:bob@192.168.1.2 >;reg-id=1
       ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
   Content-Length: 0

This SIP transaction now generates a SIP 200 OK response, as depicted
in (2) from Figure 5:

Message 2:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.2;rport=8050;branch=z9hG4bKnashds7;
     received=172.16.3.4
From: Bob <sip:bob@example.com>;tag=7F94778B653B
To: Bob <sip:bob@example.com>;tag=6AF99445E44A
Call-ID: 16CB75F21C70
CSeq: 1 REGISTER
Supported: path, outbound
Require: outbound
Contact: <sip:bob@192.168.1.2 >;reg-id=1;expires=3600
     ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
Content-Length: 0
```

The response will be sent to the address appearing in the 'received'
parameter of the SIP 'Via' header (address 172.16.3.4).  The response
will not be sent to the port deduced from the SIP 'Via' header, as
per standard SIP operation but will be sent to the value that has
been stamped in the 'rport' parameter of the SIP 'Via' header (port
8050).  For the response to successfully traverse the NAT, all of the
conventions defined in RFC 3581 [RFC3581] are to be obeyed.  Make
note of both the 'reg-id' and 'sip.instance' contact header
parameters.  They are used to establish an outbound connection tuple
as defined in [RFC5626].  The connection tuple creation is clearly
shown in Figure 5.  This ensures that any inbound request that causes
a registration lookup will result in the reuse of the connection path
established by the registration.  This removes the need to manipulate
contact header URIs to represent a globally routable address as
perceived on the public side of a NAT.

5.1.1.2.  Connection-Oriented Transport

```
                                               Registrar/
             Bob                 NAT           Edge Proxy
              |                   |                 |
              |(1) REGISTER       |                 |
              |------------------>|                 |
              |                   |                 |
              |                   |(1) REGISTER     |
              |                   |---------------->|
              |                   |                 |
              |*********************************************|
              |    Create Outbound Connection Tuple    |
              |*********************************************|
              |                   |                 |
              |                   |(2) 200 OK       |
              |                   |<----------------|
              |                   |                 |
              |(2) 200 OK         |                 |
              |<------------------|                 |
              |                   |                 |
```
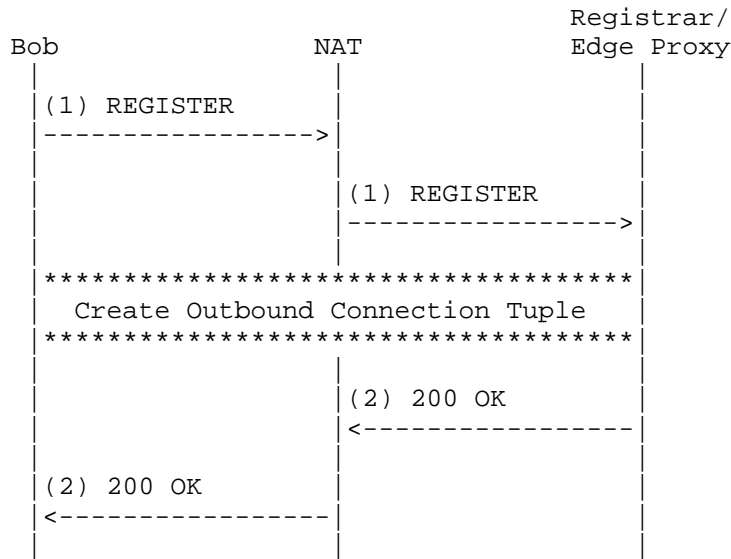
                           Figure 6

   Traversal of SIP REGISTER requests/responses using a reliable,
   connection-oriented protocol such as TCP does not require any
   additional core SIP signaling extensions, beyond the procedures
   defined in [RFC5626].  SIP responses will reuse the connection
   created for the initial REGISTER request, (1) from Figure 6:

   Message 1:

   REGISTER sip:example.com SIP/2.0
   Via: SIP/2.0/TCP 192.168.1.2;branch=z9hG4bKnashds7
   Max-Forwards: 70
   From: Bob <sip:bob@example.com>;tag=7F94778B653B
   To: Bob <sip:bob@example.com>
   Call-ID: 16CB75F21C70
   CSeq: 1 REGISTER
   Supported: path, outbound
   Contact: <sip:bob@192.168.1.2;transport=tcp>;reg-id=1
        ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
   Content-Length: 0

Message 2:

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP 192.168.1.2;branch=z9hG4bKnashds7
From: Bob <sip:bob@example.com>;tag=7F94778B653B
To: Bob <sip:bob@example.com>;tag=6AF99445E44A
Call-ID: 16CB75F21C70
CSeq: 1 REGISTER
Supported: path, outbound
Require: outbound
Contact: <sip:bob@192.168.1.2;transport=tcp>;reg-id=1;expires=3600
     ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
Content-Length: 0
```

This example was included to show the inclusion of the 'sip.instance'
contact header parameter as defined in the SIP Outbound specification
[RFC5626].  This creates an association tuple as described in the
previous example for future inbound requests directed at the newly
created registration binding with the only difference that the
association is with a TCP connection, not a UDP pinhole binding.

5.1.2.  Registration(Registrar/Edge Proxy Not Co-Located)

This section demonstrates traversal mechanisms when the Registrar
component is not co-located with the edge proxy element.  The
procedures described in this section are identical, regardless of
transport protocol, so only one example will be documented in the
form of TCP.

```
      Bob                 NAT              Edge Proxy            Registrar
       |                   |                   |                    |
       |(1) REGISTER       |                   |                    |
       |------------------>|                   |                    |
       |                   |                   |                    |
       |                   |(1) REGISTER       |                    |
       |                   |------------------>|                    |
       |                   |                   |                    |
       |                   |                   |(2) REGISTER        |
       |                   |                   |------------------->|
       |                   |                   |                    |
       |***********************************|                    |
       |   Create Outbound Connection Tuple   |                    |
       |***********************************|                    |
       |                   |                   |                    |
       |                   |                   |(3) 200 OK          |
       |                   |                   |<-------------------|
       |                   |(4)200 OK          |                    |
       |                   |<------------------|                    |
       |                   |                   |                    |
       |(4)200 OK          |                   |                    |
       |<------------------|                   |                    |
       |                   |                   |                    |
```
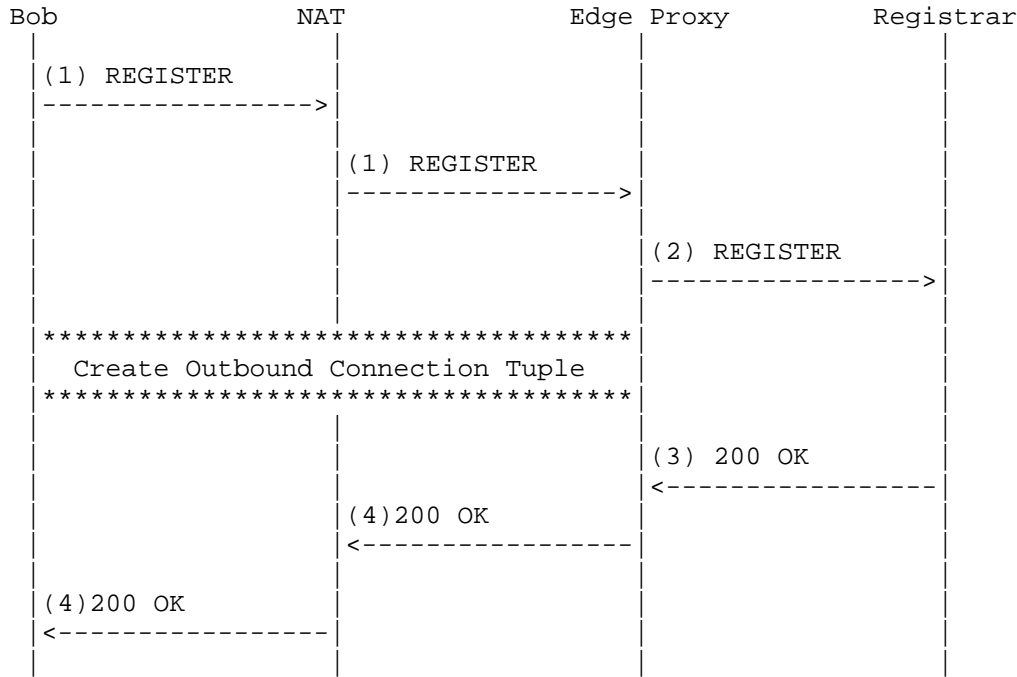
            Figure 7: Registration (Registrar/Proxy Not Co-Located)

   This scenario builds on the previous example in Section 5.1.1.2.  The
   primary difference is that the REGISTER request is routed onwards
   from a proxy server to a separated Registrar.  The important message
   to note is (1) in Figure 7.  The edge proxy, on receiving a REGISTER
   request that contains a 'sip.instance' media feature tag, forms a
   unique flow identifier token as discussed in [RFC5626].  At this
   point, the proxy server routes the SIP REGISTER message to the
   Registrar.  The proxy will create the connection tuple as described
   in SIP Outbound at the same moment as the co-located example, but for
   subsequent messages to arrive at the proxy, the proxy needs to
   indicate its need to remain in the SIP signaling path.  To achieve
   this, the proxy inserts to REGISTER message (2) a SIP 'Path'
   extension header, as defined in RFC 3327 [RFC3327].  The previously
   created flow association token is inserted in a position within the
   Path header where it can easily be retrieved at a later point when
   receiving messages to be routed to the registration binding (in this
   case the user part of the SIP URI).  The REGISTER message of (1)
   includes a SIP 'Route' header for the edge proxy.

Message 1:

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP 192.168.1.2;branch=z9hG4bKnashds7
Max-Forwards: 70
From: Bob <sip:bob@example.com>;tag=7F94778B653B
To: Bob <sip:bob@example.com>
Call-ID: 16CB75F21C70
CSeq: 1 REGISTER
Supported: path, outbound
Route: <sip:ep1.example.com;lr>
Contact: <sip:bob@192.168.1.2;transport=tcp>;reg-id=1
     ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
Content-Length: 0
```

When proxied in (2) looks as follows:

Message 2:

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP ep1.example.com;branch=z9hG4bKnuiqisi
Via: SIP/2.0/TCP 192.168.1.2;branch=z9hG4bKnashds7
Max-Forwards: 69
From: Bob <sip:bob@example.com>;tag=7F94778B653B
To: Bob <sip:bob@example.com>
Call-ID: 16CB75F21C70
CSeq: 1 REGISTER
Supported: path, outbound
Contact: <sip:bob@192.168.1.2;transport=tcp>;reg-id=1
     ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
Path: <sip:VskztcQ/S8p4WPbOnHbuyh5iJvJIW3ib@ep1.example.com;lr;ob>
Content-Length: 0
```

This REGISTER request results in the Path header being stored along
with the AOR and its associated binding at the Registrar.  The URI
contained in the Path header will be inserted as a pre-loaded SIP
'Route' header into any request that arrives at the Registrar and is
directed towards the associated AOR binding.  This all but guarantees
that all requests for the new registration will be forwarded to the
edge proxy.  In our example, the user part of the SIP 'Path' header
URI that was inserted by the edge proxy contains the unique token
identifying the flow to the client.  On receiving subsequent
requests, the edge proxy will examine the user part of the pre-loaded
SIP 'Route' header and extract the unique flow token for use in its
connection tuple comparison, as defined in the SIP Outbound
specification [RFC5626].  An example that builds on this scenario
(showing an inbound request to the AOR) is detailed in
Section 5.1.4.2 of this document.

5.1.3.  Initiating a Session

   This section covers basic SIP signaling when initiating a call from
   behind a NAT.

5.1.3.1.  UDP

   Initiating a call using UDP (the edge proxy and authoritative proxy
   functionality are co-located).

```
                                    Edge Proxy/
      Bob              NAT        Auth. Proxy           Alice
       |                |              |                  |
       |(1) INVITE      |              |                  |
       |--------------->|              |                  |
       |                |              |                  |
       |                |(1) INVITE    |                  |
       |                |------------->|                  |
       |                |              |                  |
       |                |              |(2) INVITE        |
       |                |              |----------------->|
       |                |              |                  |
       |                |              |(3)180 RINGING    |
       |                |              |<-----------------|
       |                |              |                  |
       |                |(4)180 RINGING|                  |
       |                |<-------------|                  |
       |                |              |                  |
       |(4)180 RINGING  |              |                  |
       |<---------------|              |                  |
       |                |              |                  |
       |                |              |(5)200 OK         |
       |                |              |<-----------------|
       |                |              |                  |
       |                |(6)200 OK     |                  |
       |                |<-------------|                  |
       |                |              |                  |
       |(6)200 OK       |              |                  |
       |<---------------|              |                  |
       |                |              |                  |
       |(7)ACK          |              |                  |
       |--------------->|              |                  |
       |                |              |                  |
       |                |(7)ACK        |                  |
       |                |------------->|                  |
       |                |              |                  |
       |                |              |(8) ACK           |
       |                |              |----------------->|
       |                |              |                  |
```
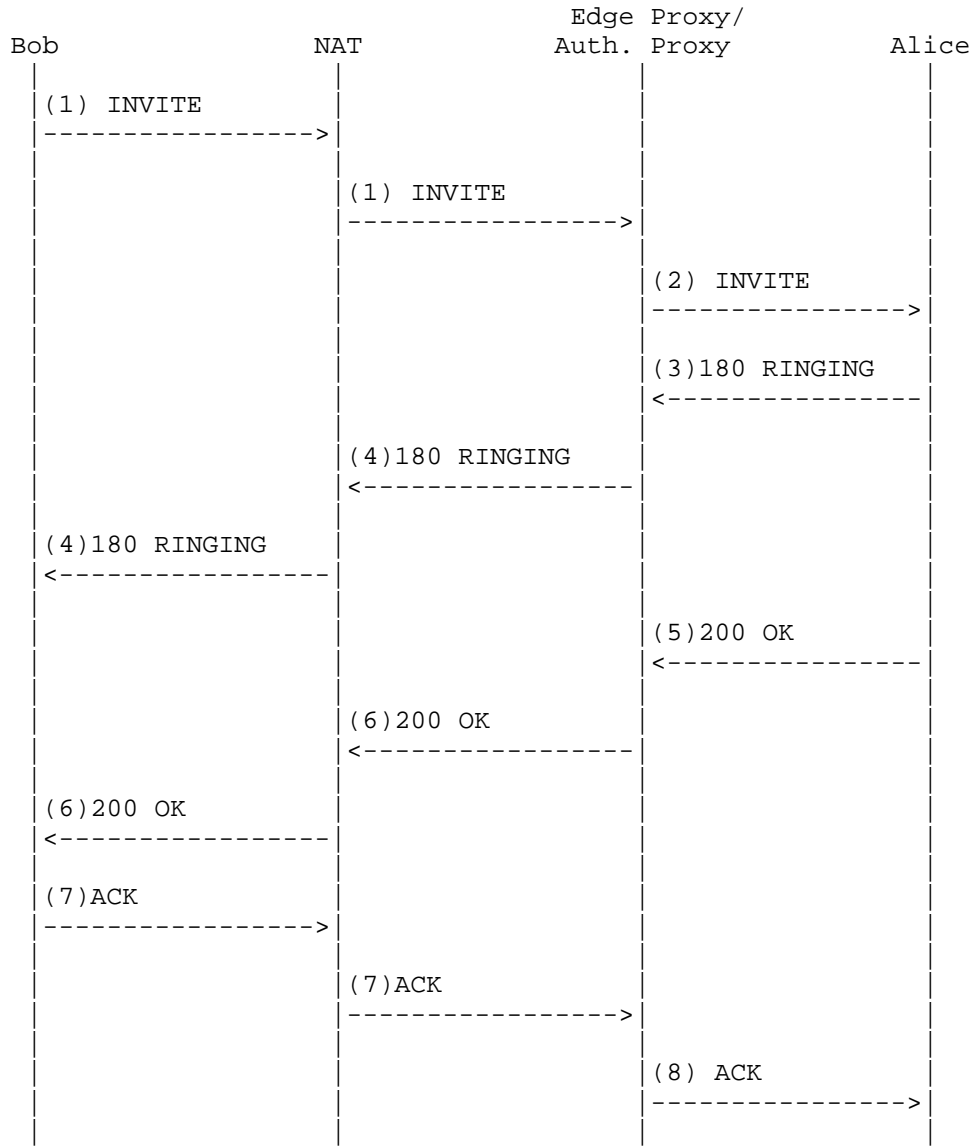
                Figure 8: Initiating a Session - UDP

The initiating client generates an INVITE request that is to be sent
through the NAT to a proxy server.  The INVITE message is represented
in Figure 8 by (1) and is as follows:

Message 1:

```
INVITE sip:alice@a.example SIP/2.0
Via: SIP/2.0/UDP 192.168.1.2;rport;branch=z9hG4bKnashds7
Max-Forwards: 70
From: Bob <sip:bob@example.com>;tag=ldw22z
To: Alice <sip:alice@a.example>
Call-ID: 95KGsk2V/Eis9LcpBYy3
CSeq: 1 INVITE
Supported: outbound
Route: <sip:ep1.example.com;lr>
Contact: <sip:bob@192.168.1.2;ob>
Content-Type: application/sdp
Content-Length: ...
```

[SDP not shown]

There are a number of points to note with this message:

1.  Firstly, as with the registration example in Section 5.1.1.1,
    responses to this request will not automatically pass back
    through a NAT, so the SIP 'Via' header 'rport' is included as
    described in the Section 4.1.1 ("Symmetric Response") and defined
    in RFC 3581 [RFC3581].

2.  Secondly, the 'ob' parameter is added to the 'Contact' header to
    ensure that all subsequent requests are sent to the same flow;
    alternatively, a Globally Routable User Agent URI (GRUU) might
    have been used.  See Section 4.3 of [RFC5626].

In (2), the proxy inserts itself in the 'Via' header, adds the
'rport' port number and the 'received' parameter in the previous
'Via' header, removes the 'Route' header, and inserts a Record-Route
with a token.

   Message 2:

```
INVITE sip:alice@172.16.1.4 SIP/2.0
Via: SIP/2.0/UDP ep1.example.com;branch=z9hG4bKnuiqisi
Via: SIP/2.0/UDP 192.168.1.2;rport=8050;branch=z9hG4bKnashds7;
     received=172.16.3.4
Max-Forwards: 69
From: Bob <sip:bob@example.com>;tag=ldw22z
To: Alice <sip:alice@a.example>
Call-ID: 95KGsk2V/Eis9LcpBYy3
CSeq: 1 INVITE
Supported: outbound
Record-Route: <sip:3yJEbr1GYZK9cPYk5Snocez6DzO7w+AX@ep1.example.com;lr>
Contact: <sip:bob@192.168.1.2;ob>
Content-Type: application/sdp
Content-Length: ...
```

   [SDP not shown]

## 5.1.3.2.  Connection-Oriented Transport

   When using a reliable transport such as TCP, the call flow and
   procedures for traversing a NAT are almost identical to those
   described in Section 5.1.3.1.  The primary difference when using
   reliable transport protocols is that symmetric response [RFC3581] is
   not required for SIP responses to traverse a NAT.  RFC 3261 [RFC3261]
   defines procedures for SIP response messages to be sent back on the
   same connection on which the request arrived.  See Section 9.5 of
   [RFC5626] for an example flow of an outgoing call.

## 5.1.4.  Receiving an Invitation to a Session

   This section details scenarios where a client behind a NAT receives
   an inbound request through a NAT.  These scenarios build on the
   previous registration scenario from Sections 5.1.1 and 5.1.2 in this
   document.

## 5.1.4.1.  Registrar/Proxy Co-Located

   The SIP signaling on the interior of the network (behind the user's
   proxy) is not impacted directly by the transport protocol, so only
   one example scenario is necessary.  The example uses UDP and follows
   on from the registration installed in the example from
   Section 5.1.1.1.

```
                                    Edge Proxy
      Bob              NAT       Auth. Proxy           Alice
       |                |            |                  |
       |**********************************************************|
       |            Registration Binding Installed in            |
       |                    Section 5.1.1.1                       |
       |**********************************************************|
       |                |            |                  |
       |                |            |(1)INVITE         |
       |                |            |<---------------   |
       |                |            |                  |
       |                |(2)INVITE   |                  |
       |                |<---------------|              |
       |                |            |                  |
       |(2)INVITE       |            |                  |
       |<---------------|            |                  |
       |                |            |                  |
       |                |            |                  |
```
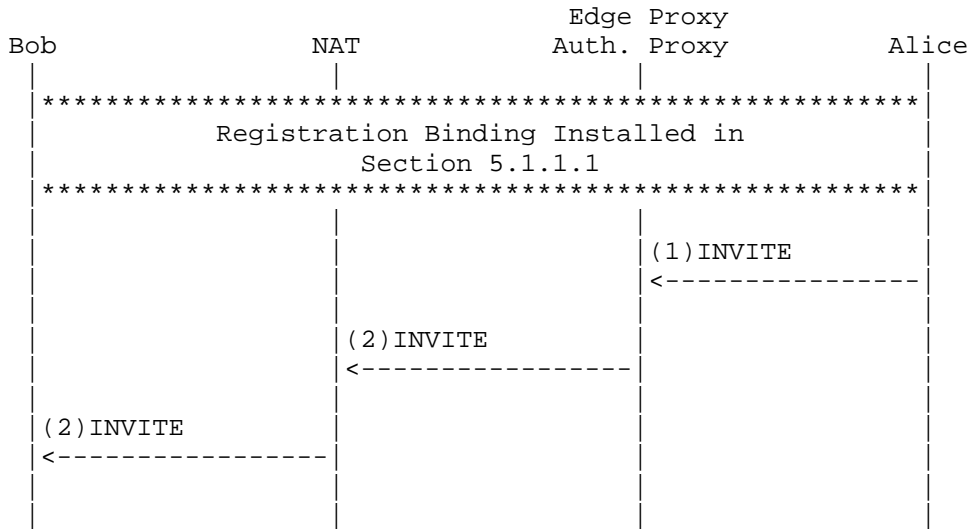
               Figure 9: Receiving an Invitation to a Session

   An INVITE request arrives at the authoritative proxy with a
   destination pointing to the AOR of that inserted in Section 5.1.1.1.
   The message is illustrated by (1) in Figure 9 and looks as follows:

   INVITE sip:bob@example.com SIP/2.0
   Via: SIP/2.0/UDP 172.16.1.4;branch=z9hG4bK74huHJ37d
   Max-Forwards: 70
   From: External Alice <sip:alice@example.com>;tag=02935
   To: Bob <sip:bob@example.com>
   Call-ID: klmvCxVWGp6MxJp2T2mb
   CSeq: 1 INVITE
   Contact: <sip:alice@172.16.1.4>
   Content-Type: application/sdp
   Content-Length: ..

   [SDP not shown]

   The INVITE request matches the registration binding previously
   installed at the Registrar and the INVITE Request-URI is rewritten to
   the selected onward address.  The proxy then examines the Request-URI
   of the INVITE and compares with its list of connection tuples.  It
   uses the incoming AOR to commence the check for associated open
   connections/mappings.  Once matched, the proxy checks to see if the
   unique instance identifier (+sip.instance) associated with the
   binding equals the same instance identifier associated with that
   connection tuple.  The request is then dispatched on the appropriate
   binding.  This is message (2) from Figure 9 and is as follows:

```
INVITE sip:bob@192.168.1.2 SIP/2.0
Via: SIP/2.0/UDP ep1.example.com;branch=z9hG4kmlds893jhsd
Via: SIP/2.0/UDP 172.16.1.4;branch=z9hG4bK74huHJ37d
Max-Forwards: 69
From: Alice <sip:alice@example.com>;tag=02935
To: client bob <sip:bob@example.com>
Call-ID: klmvCxVWGp6MxJp2T2mb
CSeq: 1 INVITE
Contact: <sip:alice@172.16.1.4>
Content-Type: application/sdp
Content-Length: ..
```

[SDP not shown]

It is a standard SIP INVITE request with no additional functionality.
The major difference is that this request will not be forwarded to
the address specified in the Request-URI, as standard SIP rules would
enforce, but will be sent on the flow associated with the
registration binding (lookup procedures in RFC 3263 [RFC3263] are
overridden by RFC 5626 [RFC5626]).  This then allows the original
connection/mapping from the initial registration process to be
reused.

5.1.4.2.  Edge Proxy/Authoritative Proxy Not Co-Located

The core SIP signaling associated with this call flow is not impacted
directly by the transport protocol, so only one example scenario is
necessary.  The example uses UDP and follows on from the registration
installed in the example from Section 5.1.2.

```
       Bob            NAT         Edge Proxy   Auth. Proxy      Alice
        |              |              |              |            |
        |**************************************************************|
        |              Registration Binding Installed in              |
        |                        Section 5.1.2                         |
        |**************************************************************|
        |              |              |              |            |
        |              |              |              |(1)INVITE   |
        |              |              |              |<------------|
        |              |              |              |            |
        |              |              |(2)INVITE     |            |
        |              |              |<------------|            |
        |              |              |              |            |
        |              |(3)INVITE     |              |            |
        |              |<------------|              |            |
        |              |              |              |            |
        |(3)INVITE     |              |              |            |
        |<------------|              |              |            |
        |              |              |              |            |
        |              |              |              |            |
```
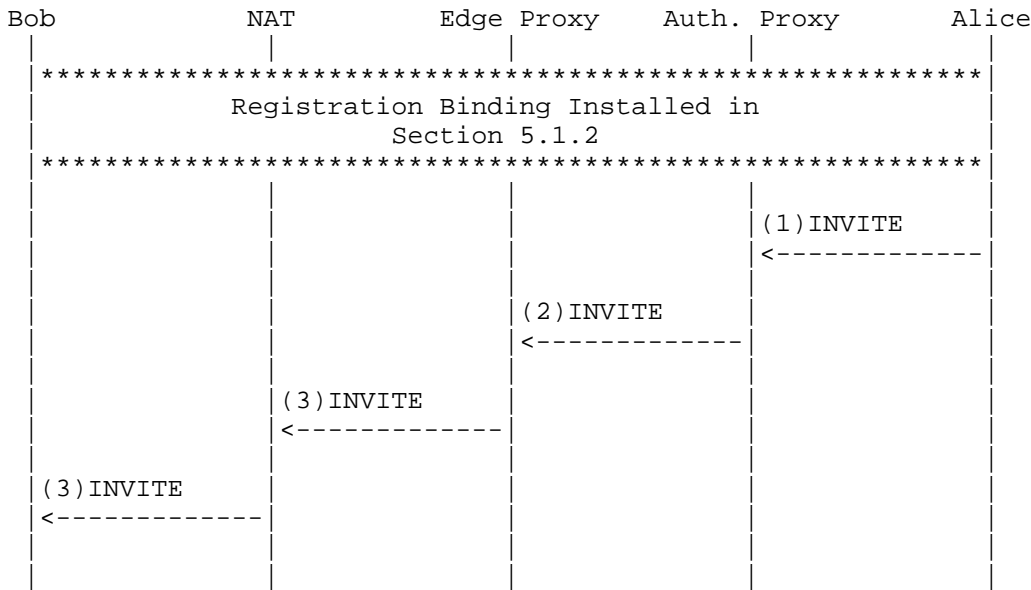
                   Figure 10: Registrar/Proxy Not Co-located

   An INVITE request arrives at the authoritative proxy with a
   destination pointing to the AOR of that inserted in Section 5.1.2.
   The message is illustrated by (1) in Figure 10 and looks as follows:

   INVITE sip:bob@example.com SIP/2.0
   Via: SIP/2.0/UDP 172.16.1.4;branch=z9hG4bK74huHJ37d
   Max-Forwards: 70
   From: Alice <sip:alice@example.com>;tag=02935
   To: Bob <sip:bob@example.com>
   Call-ID: klmvCxVWGp6MxJp2T2mb
   CSeq: 1 INVITE
   Contact: <sip:external@172.16.1.4>
   Content-Type: application/sdp
   Content-Length: ..

   [SDP not shown]

   The INVITE request matches the registration binding previously
   installed at the Registrar and the INVITE Request-URI is rewritten to
   the selected onward address.  The Registrar also identifies that a
   SIP 'Path' header was associated with the registration and pushes it
   into the INVITE request in the form of a pre-loaded SIP Route header.
   It then forwards the request on to the proxy identified in the SIP
   Route header as shown in (2) from Figure 10:

```
INVITE sip:bob@client.example.com SIP/2.0
Via: SIP/2.0/UDP proxy.example.com;branch=z9hG4bK74fmljnc
Via: SIP/2.0/UDP 172.16.1.4;branch=z9hG4bK74huHJ37d
Route: <sip:VskztcQ/S8p4WPbOnHbuyh5iJvJIW3ib@ep1.example.com;lr;ob>
Max-Forwards: 69
From: Alice <sip:alice@example.net>;tag=02935
To: Bob <sip:Bob@example.com>
Call-ID: klmvCxVWGp6MxJp2T2mb
CSeq: 1 INVITE
Contact: <sip:alice@172.16.1.4>
Content-Type: application/sdp
Content-Length: ..

[SDP not shown]
```

   The request then arrives at the outbound proxy for the client.  The
   proxy examines the Request-URI of the INVITE in conjunction with the
   flow token that it previously inserted into the user part of the
   'Path' header SIP URI (which now appears in the user part of the
   Route header in the incoming INVITE).  The proxy locates the
   appropriate flow and sends the message to the client, as shown in (3)
   from Figure 10:

```
INVITE sip:bob@192.168.1.2 SIP/2.0
Via: SIP/2.0/UDP ep1.example.com;branch=z9hG4nsi30dncmnl
Via: SIP/2.0/UDP proxy.example.com;branch=z9hG4bK74fmljnc
Via: SIP/2.0/UDP 172.16.1.4;branch=z9hG4bK74huHJ37d
Record-Route: <sip:VskztcQ/S8p4WPbOnHbuyh5iJvJIW3ib@ep1.example.com;lr>
Max-Forwards: 68
From: Alice <sip:Alice@example.net>;tag=02935
To: bob <sip:bob@example.com>
Call-ID: klmvCxVWGp6MxJp2T2mb
CSeq: 1 INVITE
Contact: <sip:alice@172.16.1.4>
Content-Type: application/sdp
Content-Length: ..

[SDP not shown]
```

   It is a standard SIP INVITE request with no additional functionality
   at the originator.  The major difference is that this request will
   not follow the address specified in the Request-URI when it reaches
   the outbound proxy, as standard SIP rules would enforce, but will be
   sent on the flow associated with the registration binding as
   indicated in the Route header (lookup procedures in RFC 3263
   [RFC3263] are overridden).  This then allows the original connection/
   mapping from the initial registration to the outbound proxy to be
   reused.

5.2.  Basic NAT Media Traversal

   This section provides example scenarios to demonstrate basic media
   traversal using the techniques outlined earlier in this document.

   In the flow diagrams, STUN messages have been annotated for
   simplicity as follows:

   o  The "Src" attribute represents the source transport address of the
      message.

   o  The "Dest" attribute represents the destination transport address
      of the message.

   o  The "Map" attribute represents the server reflexive (XOR-MAPPED-
      ADDRESS STUN attribute) transport address.

   o  The "Rel" attribute represents the relayed (RELAY-ADDRESS STUN
      attribute) transport address.

   The meaning of each STUN attribute is extensively explained in the
   core STUN [RFC5389] and TURN [RFC5766] specifications.

   A number of ICE SDP attributes have also been included in some of the
   examples.  Detailed information on individual attributes can be
   obtained from the core ICE specification [RFC5245].

   The examples also contain a mechanism for representing transport
   addresses.  It would be confusing to include representations of
   network addresses in the call flows and would make them hard to
   follow.  For this reason, network addresses will be represented using
   the following annotation.  The first component will contain the
   representation of the client responsible for the address.  For
   example, in the majority of the examples "L" (left client), "R"
   (right client), "NAT-PUB" (NAT public), "PRIV" (Private), and "STUN-
   PUB" (STUN public) are used.  To allow for multiple addresses from
   the same network element, each representation can also be followed by
   a number.  These can also be used in combination.  For example,
   "L-NAT-PUB-1" would represent a public network address of the left-
   hand side NAT while "R-NAT-PUB-1" would represent a public network
   address of the right-hand side of the NAT.  "L-PRIV-1" would
   represent a private network address of the left-hand side of the NAT
   while "R-PRIV-1" represents a private address of the right-hand side
   of the NAT.

It should also be noted that, during the examples, it might be
appropriate to signify an explicit part of a transport address.  This
is achieved by adding either the '.address' or '.port' tag on the end
of the representation -- for example, 'L-PRIV-1.address' and 'L-PRIV-
1.port'.

The use of '$' signifies variable parts in example SIP messages.

5.2.1.  Endpoint-Independent NAT

   This section demonstrates an example of a client both initiating and
   receiving calls behind an Endpoint-Independent NAT.  An example is
   included for both STUN and ICE with ICE being the RECOMMENDED
   mechanism for media traversal.

   At this time, there is no reliable test to determine if a host is
   behind an Endpoint-Independent Filtering NAT or an Endpoint-
   Independent Mapping NAT [RFC5780], and the sort of failure that
   occurs in this situation is described in Section 5.2.2.1.  For this
   reason, ICE is RECOMMENDED over the mechanism described in this
   section.

5.2.1.1.  STUN Solution

   It is possible to traverse media through an Endpoint-Independent NAT
   using STUN.  The remainder of this section provides simplified
   examples of the 'Binding Discovery' STUN as defined in [RFC5389].
   The STUN messages have been simplified and do not include 'Shared
   Secret' requests used to obtain the temporary username and password.

5.2.1.1.1.  Initiating Session

   The following example demonstrates media traversal through a NAT with
   Endpoint-Independent Mapping properties using the STUN 'Binding
   Discovery' usage.  It is assumed in this example that the STUN client
   and SIP Client are co-located on the same physical machine.  Note
   that some SIP signaling messages have been left out for simplicity.

```
        Client             NAT             STUN                 [..]
                                          Server
          |                 |                |                   |
          |(1) BIND Req     |                |                   |
          |Src=L-PRIV-1     |                |                   |
          |Dest=STUN-PUB    |                |                   |
          |---------------->|                |                   |
          |                 |                |                   |
          |                 |(2) BIND Req    |                   |
          |                 |Src=NAT-PUB-1   |                   |
          |                 |Dest=STUN-PUB   |                   |
          |                 |--------------->|                   |
          |                 |                |                   |
          |                 |(3) BIND Resp   |                   |
          |                 |<---------------|                   |
          |                 |Src=STUN-PUB    |                   |
          |                 |Dest=NAT-PUB-1  |                   |
          |                 |Map=NAT-PUB-1   |                   |
          |                 |                |                   |
          |(4) BIND Resp    |                |                   |
          |<----------------|                |                   |
          |Src=STUN-PUB     |                |                   |
          |Dest=L-PRIV-1    |                |                   |
          |Map=NAT-PUB-1    |                |                   |
          |                 |                |                   |
          |(5) BIND Req     |                |                   |
          |Src=L-PRIV-2     |                |                   |
          |Dest=STUN-PUB    |                |                   |
          |---------------->|                |                   |
          |                 |                |                   |
          |                 |(6) BIND Req    |                   |
          |                 |Src=NAT-PUB-2   |                   |
          |                 |Dest=STUN-PUB   |                   |
          |                 |--------------->|                   |
          |                 |                |                   |
          |                 |(7) BIND Resp   |                   |
          |                 |<---------------|                   |
          |                 |Src=STUN-PUB    |                   |
          |                 |Dest=NAT-PUB-2  |                   |
          |                 |Map=NAT-PUB-2   |                   |
          |                 |                |                   |
          |(8) BIND Resp    |                |                   |
          |<----------------|                |                   |
          |Src=STUN-PUB     |                |                   |
          |Dest=L-PRIV-2    |                |                   |
          |Map=NAT-PUB-2    |                |                   |
          |                 |                |                   |
```

```
 |(9)SIP INVITE      |                     |                    |
 |------------------>|                     |                    |
 |                   |                     |                    |
 |                   |(10)SIP INVITE       |                    |
 |                   |------------------------------------------>|
 |                   |                     |                    |
 |                   |                     |(11)SIP 200 OK      |
 |                   |<------------------------------------------|
 |                   |                     |                    |
 |(12)SIP 200 OK     |                     |                    |
 |<------------------|                     |                    |
 |                   |                     |                    |
 |===========================================================|
 |>>>>>>>>>>>Outgoing Media sent from L-PRIV-1>>>>>>>>>>>|
 |===========================================================|
 |                   |                     |                    |
 |===========================================================|
 |<<<<<<<<<<<Incoming Media sent to NAT-PUB-1<<<<<<<<<<<|
 |===========================================================|
 |                   |                     |                    |
 |===========================================================|
 |>>>>>>>>>>>Outgoing RTCP sent from L-PRIV-2>>>>>>>>>>>|
 |===========================================================|
 |                   |                     |                    |
 |===========================================================|
 |<<<<<<<<<<<Incoming RTCP sent to NAT-PUB-2<<<<<<<<<<<|
 |===========================================================|
 |                   |                     |                    |
 |(13)SIP ACK        |                     |                    |
 |------------------>|                     |                    |
 |                   |                     |                    |
 |                   |(14) SIP ACK         |                    |
 |                   |------------------------------------------>|
 |                   |                     |                    |
```

              Figure 11: Endpoint-Independent NAT - Initiating

   o  On deciding to initiate a SIP voice session, the client starts a
      local STUN client on the interface and port that is to be used for
      media (send/receive).  The STUN client generates a standard
      'Binding Discovery' request as indicated in (1) from Figure 11
      that also highlights the source address and port for which the
      client device wishes to obtain a mapping.  The 'Binding Discovery'
      request is sent through the NAT towards the public Internet and
      STUN server.

   o  Message (2) traverses the NAT and breaks out onto the public
      Internet towards the public STUN server.  Note that the source
      address of the 'Binding Discovery' request now represents the
      public address and port from the public side of the NAT.

   o  The STUN server receives the request and processes it
      appropriately.  This results in a successful 'Binding Discovery'
      response being generated and returned (3).  The message contains
      details of the XOR-mapped public address (contained in the STUN
      XOR-MAPPED-ADDRESS attribute) that is to be used by the
      originating client to receive media (see 'Map=NAT-PUB-1' from
      (3)).

   o  The 'Binding Discovery' response traverses back through the NAT
      using the path created by the 'Binding Discovery' request and
      presents the new XOR-mapped address to the client (4).  At this
      point, the process is repeated to obtain a second XOR-mapped
      address (as shown in (5)-(8)) for a second local address (the
      address has changed from "L-PRIV-1" to "L-PRIV-2") for an RTCP
      port.

   o  The client now constructs a SIP INVITE message (9).  Note that
      traversal of SIP is not covered in this example and is discussed
      in Section 5.1.  The INVITE request will use the addresses it has
      obtained in the previous STUN transactions to populate the SDP of
      the SIP INVITE as shown below:

      v=0
      o=test 2890844526 2890842807 IN IP4 $L-PRIV-1.address
      c=IN IP4 $NAT-PUB-1.address
      t=0 0
      m=audio $NAT-PUB-1.port RTP/AVP 0
      a=rtcp:$NAT-PUB-2.port

   o  Note that the XOR-mapped address obtained from the 'Binding
      Discovery' transactions are inserted as the connection address for
      the SDP (c=$NAT-PUB-1.address).  The Primary port for RTP is also
      inserted in the SDP (m=audio $NAT-PUB-1.port RTP/AVP 0).  Finally,
      the port gained from the additional 'Binding Discovery' is placed
      in the RTCP attribute (as discussed in Section 4.2.2) for
      traversal of RTCP (a=rtcp:$NAT-PUB-2.port).

   o  The SIP signaling then traverses the NAT and sets up the SIP
      session (9-12).  Note that the left-hand client transmits media as
      soon as the 200 OK to the INVITE arrives at the client (12).  Up
      until this point, the incoming media and RTCP to the left-hand

client will not pass through the NAT as no outbound association
has been created with the far-end client.  Two-way media
communication has now been established.

5.2.1.1.2.  Receiving Session Invitation

Receiving a session for an Endpoint-Independent NAT using the STUN
'Binding Discovery' usage is very similar to the example outlined in
Section 5.2.1.1.1.  Figure 12 illustrates the associated flow of
messages.

```
      Client                 NAT               STUN              [..]
                                              Server
         |                    |                 |  (1)SIP INVITE  |
         |                    |<----------------------------------|
         |                    |                 |                 |
         |(2) SIP INVITE      |                 |                 |
         |<---------------    |                 |                 |
         |                    |                 |                 |
         |(3) BIND Req        |                 |                 |
         |Src=L-PRIV-1        |                 |                 |
         |Dest=STUN-PUB       |                 |                 |
         |---------------->   |                 |                 |
         |                    |                 |                 |
         |                    |(4) BIND Req     |                 |
         |                    |Src=NAT-PUB-1    |                 |
         |                    |Dest=STUN-PUB    |                 |
         |                    |---------------->|                 |
         |                    |                 |                 |
         |                    |(5) BIND Resp    |                 |
         |                    |<----------------|                 |
         |                    |Src=STUN-PUB     |                 |
         |                    |Dest=NAT-PUB-1   |                 |
         |                    |Map=NAT-PUB-1    |                 |
         |                    |                 |                 |
         |(6) BIND Resp       |                 |                 |
         |<---------------    |                 |                 |
         |Src=STUN-PUB        |                 |                 |
         |Dest=L-PRIV-1       |                 |                 |
         |Map=NAT-PUB-1       |                 |                 |
         |                    |                 |                 |
         |(7) BIND Req        |                 |                 |
         |Src=L-PRIV-2        |                 |                 |
         |Dest=STUN-PUB       |                 |                 |
         |---------------->   |                 |                 |
         |                    |                 |                 |
```

```
    |                     |(8) BIND Req         |                        |
    |                     |Src=NAT-PUB-2        |                        |
    |                     |Dest=STUN-PUB        |                        |
    |                     |---------------->|                        |
    |                     |                     |                        |
    |                     |(9) BIND Resp        |                        |
    |                     |<----------------|                        |
    |                     |Src=STUN-PUB         |                        |
    |                     |Dest=NAT-PUB-2       |                        |
    |                     |Map=NAT-PUB-2        |                        |
    |                     |                     |                        |
    |(10) BIND Resp       |                     |                        |
    |<----------------|                     |                        |
    |Src=STUN-PUB         |                     |                        |
    |Dest=L-PRIV-2        |                     |                        |
    |Map=NAT-PUB-2        |                     |                        |
    |                     |                     |                        |
    |(11)SIP 200 OK       |                     |                        |
    |---------------->|                     |                        |
    |                     |(12)SIP 200 OK       |                        |
    |                     |------------------------------------>|
    |                     |                     |                        |
    |=======================================================|
    |>>>>>>>>>>>>Outgoing Media sent from L-PRIV-1>>>>>>>>>>>|
    |=======================================================|
    |                     |                     |                        |
    |=======================================================|
    |<<<<<<<<<<<<Incoming Media sent to L-PRIV-1<<<<<<<<<<<<|
    |=======================================================|
    |                     |                     |                        |
    |=======================================================|
    |>>>>>>>>>>>>Outgoing RTCP sent from L-PRIV-2>>>>>>>>>>>>|
    |=======================================================|
    |                     |                     |                        |
    |=======================================================|
    |<<<<<<<<<<<<Incoming RTCP sent to L-PRIV-2<<<<<<<<<<<<<|
    |=======================================================|
    |                     |                     |                        |
    |                     |                     |(13)SIP ACK             |
    |                     |<------------------------------------|
    |                     |                     |                        |
    |(14)SIP ACK          |                     |                        |
    |<----------------|                     |                        |
    |                     |                     |                        |
```

                Figure 12: Endpoint-Independent NAT - Receiving

o  On receiving an invitation to a SIP voice session (SIP INVITE
   request), the User Agent starts a local STUN client on the
   appropriate port on which it is to receive media.  The STUN client
   generates a standard 'Binding Discovery' request as indicated in
   (3) from Figure 12 that also highlights the source address and
   port for which the client device wishes to obtain a mapping.  The
   'Binding Discovery' request is sent through the NAT towards the
   public Internet and STUN server.

o  'Binding Discovery' message (4) traverses the NAT and breaks out
   onto the public Internet towards the public STUN server.  Note
   that the source address of the STUN requests now represents the
   public address and port from the public side of the NAT.

o  The STUN server receives the request and processes it
   appropriately.  This results in a successful 'Binding Discovery'
   response being generated and returned (5).  The message contains
   details of the mapped public address (contained in the STUN XOR-
   MAPPED-ADDRESS attribute) that is to be used by the originating
   client to receive media (see 'Map=NAT-PUB-1' from (5)).

o  The 'Binding Discovery' response traverses back through the NAT
   using the path created by the outgoing 'Binding Discovery' request
   and presents the new XOR-mapped address to the client (6).  At
   this point, the process is repeated to obtain a second XOR-mapped
   address (as shown in (7)-(10)) for a second local address (local
   port has now changed and is represented by L-PRIV-2 in (7)) for an
   RTCP port.

o  The client now constructs a SIP 200 OK message (11) in response to
   the original SIP INVITE requests.  Note that traversal of SIP is
   not covered in this example and is discussed in Section 5.1.  SIP
   Provisional responses are also left out for simplicity.  The 200
   OK response will use the addresses it has obtained in the previous
   STUN transactions to populate the SDP of the SIP 200 OK as shown
   below:

   v=0
   o=test 2890844526 2890842807 IN IP4 $L-PRIV-1.address
   c=IN IP4 $NAT-PUB-1.address
   t=0 0
   m=audio $NAT-PUB-1.port RTP/AVP 0
   a=rtcp:$NAT-PUB-2.port

o  Note that the XOR-mapped address obtained from the initial
   'Binding Discovery' transaction is inserted as the connection
   address for the SDP (c=NAT-PUB-1.address).  The Primary port for
   RTP is also inserted in the SDP (m=audio NAT-PUB-1.port RTP/AVP

0).  Finally, the port gained from the second 'Binding Discovery'
is placed in the RTCP attribute (as discussed in Section 4.2.2)
for traversal of RTCP (a=rtcp:NAT-PUB-2.port).

o  The SIP signaling then traverses the NAT and sets up the SIP
session (11-14).  Note that the left-hand client transmits media
as soon as the 200 OK to the INVITE is sent to the User Agent
Client (UAC) (11).  Up until this point, the incoming media from
the right-hand client will not pass through the NAT as no outbound
association has been created with the far-end client.  Two-way
media communication has now been established.

## 5.2.1.2.  ICE Solution

The preferred solution for media traversal of NAT is using ICE, as
described in Section 4.2.3.3, regardless of the NAT type.  The
following examples illustrate the traversal of an Endpoint-
Independent NAT when initiating the session.  The example only covers
ICE in association with the 'Binding Discovery' and TURN.  It is
worth noting that the TURN server provides both STUN functions (to
learn your public mapping) and TURN functions (media relaying).  It
is also worth noting that in the example described in
Section 5.2.1.2.1, both SIP clients L and R are contacting the same
TURN server.  This is not necessary for ICE, STUN, TURN to function;
all that is necessary is that the STUN and TURN server(s) be in the
same addressing domain that is accessible on the Internet.

## 5.2.1.2.1.  Initiating Session

The following example demonstrates an initiating traversal through an
Endpoint-Independent NAT using ICE.

```
     L              NAT            STUN            NAT              R
                                  Server
     |               |              |              |               |
     |(1) Alloc Req  |              |              |               |
     |Src=L-PRIV-1   |              |              |               |
     |Dest=TURN-PUB-1|              |              |               |
     |-------------->|              |              |               |
     |               |              |              |               |
     |               |(2) Alloc Req |              |               |
     |               |Src=L-NAT-PUB-1              |               |
     |               |Dest=TURN-PUB-1              |               |
     |               |------------->|              |               |
     |               |              |              |               |
     |               |(3) Alloc Resp|              |               |
     |               |<-------------|              |               |
     |               |Src=TURN-PUB-1|              |               |
     |               |Dest=L-NAT-PUB-1             |               |
     |               |Map=L-NAT-PUB-1             |               |
     |               |Rel=TURN-PUB-2|              |               |
     |               |              |              |               |
     |(4) Alloc Resp |              |              |               |
     |<--------------|              |              |               |
     |Src=TURN-PUB-1 |              |              |               |
     |Dest=L-PRIV-1  |              |              |               |
     |Map=L-NAT-PUB-1|              |              |               |
     |Rel=TURN-PUB-2 |              |              |               |
     |               |              |              |               |
     |(5) Alloc Req  |              |              |               |
     |Src=L-PRIV-2   |              |              |               |
     |Dest=TURN-PUB-1|              |              |               |
     |-------------->|              |              |               |
     |               |              |              |               |
     |               |(6) Alloc Req |              |               |
     |               |Src=L-NAT-PUB-2              |               |
     |               |Dest=TURN-PUB-1              |               |
     |               |------------->|              |               |
     |               |              |              |               |
     |               |(7) Alloc Resp|              |               |
     |               |<-------------|              |               |
     |               |Src=TURN-PUB-1|              |               |
     |               |Dest=NAT-PUB-2|              |               |
     |               |Map=NAT-PUB-2 |              |               |
     |               |Rel=TURN-PUB-3|              |               |
     |               |              |              |               |
```

```
|(8) Alloc Resp |               |               |               |
|<--------------|               |               |               |
|Src=TURN-PUB-1 |               |               |               |
|Dest=L-PRIV-2  |               |               |               |
|Map=L-NAT-PUB-2|               |               |               |
|Rel=TURN-PUB-3 |               |               |               |
|               |               |               |               |
|(9) SIP INVITE |               |               |               |
|---------------------------------------------------->|         |
|               |               |               |               |
|               |               |               |(10) SIP INVITE|
|               |               |               |-------------->|
|               |               |               |               |
|               |               |               |(11) Alloc Req |
|               |               |               |<--------------|
|               |               |               |Src=R-PRIV-1   |
|               |               |               |Dest=TURN-PUB-1|
|               |               |               |               |
|               |               |(12) Alloc Req |               |
|               |               |<--------------|               |
|               |               |Src=R-NAT-PUB-1|               |
|               |               |Dest=TURN-PUB-1|               |
|               |               |               |               |
|               |               |(13) Alloc Res |               |
|               |               |-------------->|               |
|               |               |Src=TURN-PUB-1 |               |
|               |               |Dest=R-NAT-PUB-1               |
|               |               |Map=R-NAT-PUB-1|               |
|               |               |Rel=TURN-PUB-4 |               |
|               |               |               |               |
|               |               |               |(14) Alloc Res |
|               |               |               |-------------->|
|               |               |               |Src=TURN-PUB-1 |
|               |               |               |Dest=R-PRIV-1  |
|               |               |               |Map=R-NAT-PUB-1|
|               |               |               |Rel=TURN-PUB-4 |
|               |               |               |               |
|               |               |               |(15) Alloc Req |
|               |               |               |<--------------|
|               |               |               |Src=R-PRIV-2   |
|               |               |               |Dest=TURN-PUB-1|
|               |               |               |               |
|               |               |(16) Alloc Req |               |
|               |               |<--------------|               |
|               |               |Src=R-NAT-PUB-2|               |
|               |               |Dest=TURN-PUB-1|               |
|               |               |               |               |
```

```
|                 |                 |(17) Alloc Res   |                 |
|                 |                 |---------------->|                 |
|                 |                 |Src=TURN-PUB-1   |                 |
|                 |                 |Dest=R-NAT-PUB-2 |                 |
|                 |                 |Map=R-NAT-PUB-2  |                 |
|                 |                 |Rel=TURN-PUB-5   |                 |
|                 |                 |                 |                 |
|                 |                 |                 |(18) Alloc Res   |
|                 |                 |                 |---------------->|
|                 |                 |                 |Src=TURN-PUB-1   |
|                 |                 |                 |Dest=R-PRIV-2    |
|                 |                 |                 |Map=R-NAT-PUB-2  |
|                 |                 |                 |Rel=TURN-PUB-5   |
|                 |                 |                 |                 |
|                 |                 |                 |(19) SIP 200 OK  |
|                 |     <-------------------------------------------------
|                 |                 |                 |                 |
|(20) SIP 200 OK  |                 |                 |                 |
|<--------------  |                 |                 |                 |
|                 |                 |                 |                 |
|(21) SIP ACK     |                 |                 |                 |
|--------------------------------------------------->|                 |
|                 |                 |                 |                 |
|                 |                 |                 |(22) SIP ACK     |
|                 |                 |                 |---------------->|
|                 |                 |                 |                 |
|(23) Bind Req    |                 |                 |                 |
|---------------------->x          |                 |                 |
|Src=L-PRIV-1     |                 |                 |                 |
|Dest=R-PRIV-1    |                 |                 |                 |
|                 |                 |                 |                 |
|(24) Bind Req    |                 |                 |                 |
|--------------> |                 |                 |                 |
|Src=L-PRIV-1     |                 |                 |                 |
|Dest=R-NAT-PUB-1 |                 |                 |                 |
|                 |(25) Bind Req    |                 |                 |
|                 |------------------------------>|   |                 |
|                 |Src=L-NAT-PUB-1  |                 |                 |
|                 |Dest=R-NAT-PUB-1 |                 |                 |
|                 |                 |                 |                 |
|                 |                 |                 |(26) Bind Req    |
|                 |                 |                 |---------------->|
|                 |                 |                 |Src=L-NAT-PUB-1  |
|                 |                 |                 |Dest=R-PRIV-1    |
|                 |                 |                 |                 |
```

```
|                 |                 |                 |(27) Bind Res    |
|                 |                 |                 |<----------------|
|                 |                 |                 |Src=R-PRIV-1     |
|                 |                 |                 |Dest=L-NAT-PUB-1 |
|                 |                 |                 |Map=L-NAT-PUB-1  |
|                 |                 |                 |                 |
|                 |                 |(28) Bind Res    |                 |
|                 |                 |<--------------------------------|   |
|                 |                 |Src=R-NAT-PUB-1  |                 |
|                 |                 |Dest=L-NAT-PUB-1 |                 |
|                 |                 |Map=L-NAT-PUB-1  |                 |
|                 |                 |                 |                 |
|(29) Bind Res    |                 |                 |                 |
|<----------------|                 |                 |                 |
|Src=R-NAT-PUB-1  |                 |                 |                 |
|Dest=L-PRIV-1    |                 |                 |                 |
|Map=L-NAT-PUB-1  |                 |                 |                 |
|                 |                 |                 |                 |
|=================================================================|
|>>>>>>>>>>>>>>>>>Outgoing RTP sent from L-PRIV-1 >>>>>>>>>>>>>>>>>|
|=================================================================|
|                 |                 |                 |                 |
|                 |                 |                 |(30) Bind Req    |
|                 |                 |              x<--------------------|
|                 |                 |                 |Src=R-PRIV-1     |
|                 |                 |                 |Dest=L-PRIV-1    |
|                 |                 |                 |                 |
|                 |                 |                 |(31) Bind Req    |
|                 |                 |                 |<----------------|
|                 |                 |                 |Src=R-PRIV-1     |
|                 |                 |                 |Dest=L-NAT-PUB-1 |
|                 |                 |                 |                 |
|                 |                 |(32) Bind Req    |                 |
|                 |                 |<--------------------------------|   |
|                 |                 |Src=R-NAT-PUB-1  |                 |
|                 |                 |Dest=L-NAT-PUB-1 |                 |
|                 |                 |                 |                 |
|(33) Bind Req    |                 |                 |                 |
|<----------------|                 |                 |                 |
|Src=R-NAT-PUB-1  |                 |                 |                 |
|Dest=L-PRIV-1    |                 |                 |                 |
|                 |                 |                 |                 |
|(34) Bind Res    |                 |                 |                 |
|---------------->|                 |                 |                 |
|Src=L-PRIV-1     |                 |                 |                 |
|Dest=R-NAT-PUB-1 |                 |                 |                 |
|Map=R-NAT-PUB-1  |                 |                 |                 |
|                 |                 |                 |                 |
```

```
|                 |(35) Bind Res  |                  |                 |
|                 |----------------------------------->|                 |
|                 |Src=L-NAT-PUB-1|                  |                 |
|                 |Dest=R-NAT-PUB-1                   |                 |
|                 |Map=R-NAT-PUB-1|                  |                 |
|                 |               |                  |                 |
|                 |               |                  |(36) Bind Res    |
|                 |               |                  |--------------->|
|                 |               |                  |Src=L-NAT-PUB-1 |
|                 |               |                  |Dest=R-PRIV-1   |
|                 |               |                  |Map=R-NAT-PUB-1 |
|                 |               |                  |                 |
|==================================================================|
|<<<<<<<<<<<<<<<<<Outgoing RTP sent from R-PRIV-1 <<<<<<<<<<<<<<<<<|
|==================================================================|
|(37) Bind Req    |               |                  |                 |
|--------------->|               |                  |                 |
|Src=L-PRIV-1    |               |                  |                 |
|Dest=R-NAT-PUB-1|               |                  |                 |
|USE-CANDIDATE   |               |                  |                 |
|                 |               |                  |                 |
|                 |(38) Bind Req  |                  |                 |
|                 |----------------------------------->|                 |
|                 |Src=L-NAT-PUB-1|                  |                 |
|                 |Dest=R-NAT-PUB-1                   |                 |
|                 |USE-CANDIDATE  |                  |                 |
|                 |               |                  |                 |
|                 |               |                  |(39) Bind Req    |
|                 |               |                  |--------------->|
|                 |               |                  |Src=L-NAT-PUB-1 |
|                 |               |                  |Dest=R-PRIV-1   |
|                 |               |                  |USE-CANDIDATE   |
|                 |               |                  |                 |
|                 |               |                  |(40) Bind Res    |
|                 |               |                  |<---------------|
|                 |               |                  |Src=R-PRIV-1    |
|                 |               |                  |Dest=L-NAT-PUB-1|
|                 |               |                  |Map=L-NAT-PUB-1 |
|                 |               |                  |                 |
|                 |               |(41) Bind Res     |                 |
|                 |<-----------------------------------|                 |
|                 |               |Src=R-NAT-PUB-1   |                 |
|                 |               |Dest=L-NAT-PUB-1  |                 |
|                 |               |Map=L-NAT-PUB-1   |                 |
|                 |               |                  |                 |
```

```
|(42) Bind Res   |                |                |                |
|<---------------|                |                |                |
|Src=R-NAT-PUB-1 |                |                |                |
|Dest=L-PRIV-1   |                |                |                |
|Map=L-NAT-PUB-1 |                |                |                |
|                |                |                |                |
|(43) Bind Req   |                |                |                |
|--------------->|                |                |                |
|Src=L-PRIV-2    |                |                |                |
|Dest=R-NAT-PUB-2|                |                |                |
|                |                |                |                |
|                |(44) Bind Req   |                |                |
|                |------------------------------->|                |
|                |Src=L-NAT-PUB-2 |                |                |
|                |Dest=R-NAT-PUB-2|                |                |
|                |                |                |                |
|                |                |                |(45) Bind Req   |
|                |                |                |--------------->|
|                |                |                |Src=L-NAT-PUB-2 |
|                |                |                |Dest=R-PRIV-2   |
|                |                |                |                |
|                |                |                |(46) Bind Res   |
|                |                |                |<---------------|
|                |                |                |Src=R-PRIV-2    |
|                |                |                |Dest=L-NAT-PUB-2|
|                |                |                |Map=L-NAT-PUB-2 |
|                |                |                |                |
|                |                |(47) Bind Res   |                |
|                |<-------------------------------|                |
|                |                |Src=R-NAT-PUB-2 |                |
|                |                |Dest=L-NAT-PUB-2|                |
|                |                |Map=L-NAT-PUB-2 |                |
|                |                |                |                |
|(48) Bind Res   |                |                |                |
|<---------------|                |                |                |
|Src=R-NAT-PUB-2 |                |                |                |
|Dest=L-PRIV-2   |                |                |                |
|Map=L-NAT-PUB-2 |                |                |                |
|                |                |                |                |
|================================================================|
|>>>>>>>>>>>>>>>>>Outgoing RTCP sent from L-PRIV-2 >>>>>>>>>>>>>>>>|
|================================================================|
|                |                |                |                |
|                |                |                |(49) Bind Req   |
|                |                |                |<---------------|
|                |                |                |Src=R-PRIV-2    |
|                |                |                |Dest=L-NAT-PUB-2|
|                |                |                |                |
```

```
|                     |                     |(50) Bind Req        |                     |
|                     |<--------------------------------|                     |
|                     |                     |Src=R-NAT-PUB-2      |                     |
|                     |                     |Dest=L-NAT-PUB-2     |                     |
|                     |                     |                     |                     |
|(51) Bind Req        |                     |                     |                     |
|<--------------|     |                     |                     |                     |
|Src=R-NAT-PUB-2|     |                     |                     |                     |
|Dest=L-PRIV-2  |     |                     |                     |                     |
|                     |                     |                     |                     |
|(52) Bind Res        |                     |                     |                     |
|--------------->     |                     |                     |                     |
|Src=L-PRIV-2         |                     |                     |                     |
|Dest=R-NAT-PUB-2|    |                     |                     |                     |
|Map=R-NAT-PUB-2|     |                     |                     |                     |
|                     |                     |                     |                     |
|                     |(53) Bind Res        |                     |                     |
|                     |--------------------------------->|                     |
|                     |Src=L-NAT-PUB-2 |    |                     |                     |
|                     |Dest=R-NAT-PUB-2|    |                     |                     |
|                     |Map=R-NAT-PUB-2 |    |                     |                     |
|                     |                     |                     |                     |
|                     |                     |                     |(54) Bind Res        |
|                     |                     |                     |--------------->     |
|                     |                     |                     |Src=L-NAT-PUB-2      |
|                     |                     |                     |Dest=R-PRIV-2        |
|                     |                     |                     |Map=R-NAT-PUB-2      |
|                     |                     |                     |                     |
|=====================================================================|
|<<<<<<<<<<<<<<<<Outgoing RTCP sent from R-PRIV-2<<<<<<<<<<<<<<<<<|
|=====================================================================|
|(55) Bind Req        |                     |                     |                     |
|--------------->     |                     |                     |                     |
|Src=L-PRIV-2         |                     |                     |                     |
|Dest=R-NAT-PUB-2|    |                     |                     |                     |
|USE-CANDIDATE        |                     |                     |                     |
|                     |                     |                     |                     |
|                     |(56) Bind Req        |                     |                     |
|                     |--------------------------------->|                     |
|                     |Src=L-NAT-PUB-2 |    |                     |                     |
|                     |Dest=R-NAT-PUB-2|    |                     |                     |
|                     |USE-CANDIDATE        |                     |                     |
|                     |                     |                     |                     |
|                     |                     |                     |(57) Bind Req        |
|                     |                     |                     |--------------->     |
|                     |                     |                     |Src=L-NAT-PUB-2      |
|                     |                     |                     |Dest=R-PRIV-2        |
|                     |                     |                     |USE-CANDIDATE        |
```

```
|                |             |             |                 |       |
|                |             |             | (58) Bind Res   |       |
|                |             |             |<----------------|       |
|                |             |             | Src=R-PRIV-2    |       |
|                |             |             | Dest=L-NAT-PUB-2|       |
|                |             |             | Map=L-NAT-PUB-2 |       |
|                |             |             |                 |       |
|                |             | (59) Bind Res                 |       |
|                |             |<------------------------------|       |
|                |             | Src=R-NAT-PUB-2               |       |
|                |             | Dest=L-NAT-PUB-2              |       |
|                |             | Map=L-NAT-PUB-2               |       |
|                |             |             |                 |       |
| (60) Bind Res  |             |             |                 |       |
|<---------------|             |             |                 |       |
| Src=R-NAT-PUB-2|             |             |                 |       |
| Dest=L-PRIV-2  |             |             |                 |       |
| Map=L-NAT-PUB-2|             |             |                 |       |
|                |             |             |                 |       |
| (61) SIP INVITE|             |             |                 |       |
|------------------------------------------------------->|            |
|                |             |             |                 |       |
|                |             |             | (62) SIP INVITE |       |
|                |             |             |--------------->|        |
|                |             |             |                 |       |
|                |             |             | (63) SIP 200 OK |       |
|                |             |<------------------------------|       |
|                |             |             |                 |       |
| (64) SIP 200 OK|             |             |                 |       |
|<---------------|             |             |                 |       |
|                |             |             |                 |       |
| (65) SIP ACK   |             |             |                 |       |
|------------------------------------------------------->|            |
|                |             |             |                 |       |
|                |             |             | (66) SIP ACK    |       |
|                |             |             |--------------->|        |
|                |             |             |                 |       |
```

Figure 13: Endpoint-Independent NAT with ICE

o  On deciding to initiate a SIP voice session, the SIP client L
   starts a local STUN client.  The STUN client generates a TURN
   Allocate request as indicated in (1) from Figure 13 that also
   highlights the source address and port combination for which the
   client device wishes to obtain a mapping.  The Allocate request is
   sent through the NAT towards the public Internet.

o  The Allocate message (2) traverses the NAT to the public Internet
   towards the public TURN server.  Note that the source address of
   the Allocate request now represents the public address and port
   from the public side of the NAT (L-NAT-PUB-1).

o  The TURN server receives the Allocate request and processes it
   appropriately.  This results in a successful Allocate response
   being generated and returned (3).  The message contains details of
   the server reflexive address that is to be used by the originating
   client to receive media (see 'Map=L-NAT-PUB-1') from (3)).  It
   also contains an appropriate TURN-relayed address that can be used
   at the STUN server (see 'Rel=TURN-PUB-2').

o  The Allocate response traverses back through the NAT using the
   binding created by the initial Allocate request and presents the
   new mapped address to the client (4).  The process is repeated and
   a second STUN derived set of addresses is obtained, as illustrated
   in (5)-(8) in Figure 13.  At this point, the User Agent behind the
   NAT has pairs of derived external server reflexive and relayed
   representations.  The client can also gather IP addresses and
   ports via other mechanisms (e.g., NAT-PMP [NAT-PMP], UPnP IGD
   [UPnP-IGD]) or similar.

o  The client now constructs a SIP INVITE message (9).  The INVITE
   request will use the addresses it has obtained in the previous
   STUN/TURN interactions to populate the SDP of the SIP INVITE.
   This should be carried out in accordance with the semantics
   defined in the ICE specification [RFC5245], as shown below in
   Figure 14:

```
v=0
o=test 2890844526 2890842807 IN IP4 $L-PRIV-1
c=IN IP4 $L-PRIV-1.address
t=0 0
a=ice-pwd:$LPASS
a=ice-ufrag:$LUNAME
m=audio $L-PRIV-1.port RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=rtcp:$L-PRIV-2.port
a=candidate:$L1 1 UDP 2130706431 $L-PRIV-1.address $L-PRIV-1.port
            typ host
a=candidate:$L1 2 UDP 2130706430 $L-PRIV-2.address $L-PRIV-2.port
            typ host
a=candidate:$L2 1 UDP 1694498815 $L-NAT-PUB-1.address $L-NAT-PUB-1.port
            typ srflx raddr $L-PRIV-1.address rport $L-PRIV-1.port
a=candidate:$L2 2 UDP 1694498814 $L-NAT-PUB-2.address $L-NAT-PUB-2.port
            typ srflx raddr $L-PRIV-1.address rport $L-PRIV-2.port
a=candidate:$L3 1 UDP 16777215 $STUN-PUB-2.address $STUN-PUB-2.port
            typ relay raddr $L-PRIV-1.address rport $L-PRIV-1.port
a=candidate:$L3 2 UDP 16777214 $STUN-PUB-3.address $STUN-PUB-3.port
            typ relay raddr $L-PRIV-1.address rport $L-PRIV-2.port
```

                       Figure 14: ICE SDP Offer

   o  The SDP has been constructed to include all the available
      candidates that have been assembled.  The first set of candidates
      (as identified by Foundation $L1) contains two local addresses
      that have the highest priority.  They are also encoded into the
      connection (c=) and media (m=) lines of the SDP.  The second set
      of candidates, as identified by Foundation $L2, contains the two
      server reflexive addresses obtained from the STUN server for both
      RTP and RTCP traffic (identified by candidate-id $L2).  This entry
      has been given a priority lower than the pair $L1 by the client.
      The third and final set of candidates represents the relayed
      addresses (as identified by $L3) obtained from the STUN server.
      This pair has the lowest priority and will be used as a last
      resort if both $L1 and $L2 fail.

   o  The SIP signaling then traverses the NAT and sets up the SIP
      session (9)-(10).  On advertising a candidate address, the client
      should have a local STUN server running on each advertised
      candidate address.  This is for the purpose of responding to
      incoming STUN connectivity checks.

   o  On receiving the SIP INVITE request (10) client R also starts
      local STUN servers on appropriate address/port combinations and
      gathers potential candidate addresses to be encoded into the SDP
      (as the originating client did).  Steps (11-18) involve client R

carrying out the same steps as client L. This involves obtaining
local, server reflexive, and relayed addresses.  Client R is now
ready to generate an appropriate answer in the SIP 200 OK message
(19).  The example answer follows in Figure 15:

```
v=0
o=test 3890844516 3890842803 IN IP4 $R-PRIV-1
c=IN IP4 $R-PRIV-1.address
t=0 0
a=ice-pwd:$RPASS
m=audio $R-PRIV-1.port RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=rtcp:$R-PRIV-2.port
a=candidate:$L1 1 UDP 2130706431 $R-PRIV-1.address $R-PRIV-1.port
          typ host
a=candidate:$L1 2 UDP 2130706430 $R-PRIV-2.address $R-PRIV-2.port
          typ host
a=candidate:$L2 1 UDP 1694498815 $R-NAT-PUB-1.address $R-NAT-PUB-1.port
          typ srflx raddr $R-PRIV-1.address rport $R-PRIV-1.port
a=candidate:$L2 2 UDP 1694498814 $R-NAT-PUB-2.address $R-NAT-PUB-2.port
          typ srflx raddr $R-PRIV-1.address rport $R-PRIV-1.port
a=candidate:$L3 1 UDP 16777215 $STUN-PUB-2.address $STUN-PUB-4.port
          typ relay raddr $R-PRIV-1.address rport $R-PRIV-1.port
a=candidate:$L3 2 UDP 16777214 $STUN-PUB-3.address $STUN-PUB-5.port
          typ relay raddr $R-PRIV-1.address rport $R-PRIV-1.port
```

Figure 15: ICE SDP Answer

o  The two clients have now exchanged SDP using offer/answer and can
   now continue with the ICE processing -- User Agent L assuming the
   role controlling agent, as specified by ICE.  The clients are now
   required to form their Candidate check lists to determine which
   will be used for the media streams.  In this example, User Agent
   L's Foundation 1 is paired with User Agent R's Foundation 1, User
   Agent L's Foundation 2 is paired with User Agent R's Foundation 2,
   and finally User Agent L's Foundation 3 is paired with User Agent
   R's Foundation 3.  User Agents L and R now have a complete
   candidate checklist.  Both clients now use the algorithm provided
   in ICE to determine candidate pair priorities and sort into a list
   of decreasing priorities.  In this example, both User Agents L and
   R will have lists that firstly specify the host address
   (Foundation $L1), then the server reflexive address (Foundation
   $L2), and lastly the relayed address (Foundation $L3).  All
   candidate pairs have an associate state as specified in ICE.  At
   this stage, all of the candidate pairs for User Agents L and R are
   initialized to the 'Frozen' state.  The User Agents then scan the
   list and move the candidates to the 'Waiting' state.  At this
   point, both clients will periodically, starting with the highest

candidate pair priority, work their way down the list issuing STUN
checks from the local candidate to the remote candidate (of the
candidate pair).  As a STUN check is attempted from each local
candidate in the list, the candidate pair state transitions to
'In-Progress'.  As illustrated in (23), client L constructs a STUN
connectivity check in an attempt to validate the remote candidate
address received in the SDP of the 200 OK (20) for the highest
priority in the checklist.  As a private address was specified in
the active address in the SDP, the STUN connectivity check fails
to reach its destination causing a STUN failure.  Client L
transitions the state for this candidate pair to 'Failed'.  In the
meantime, client L is attempting a STUN connectivity check for the
second candidate pair in the returned SDP with the second highest
priority (24).  As can be seen from messages (24) to (29), the
STUN Bind request is successful and returns a positive outcome for
the connectivity check.  Client L is now free to send media to the
peer using the candidate pair.  Immediately after sending its 200
OK, client R also carries out the same set of binding requests.
It firstly (in parallel) tries to contact the active address
contained in the SDP (30) which results in failure.

o  In the meantime, a successful response to a STUN connectivity
   check by User Agent R (27) results in a tentative check in the
   reverse direction -- this is illustrated by messages (31) to (36).
   Once this check has succeeded, User Agent R can transition the
   state of the appropriate candidate to 'Succeeded', and media can
   be sent (RTP).  The previously (31-36) described check confirm on
   both sides (User Agents L and R) that connectivity can be achieved
   using the appropriate candidate pair.  User Agent L, as the
   controlling client now sends another connectivity check for the
   candidate pair, this time including the 'USE-CANDIDATE' attribute
   as specified in ICE to signal the chosen candidate.  This exchange
   is illustrated in messages (37) to (42).

o  As part of the process in this example, both L and R will now
   complete the same connectivity checks for part 2 of the component
   named for the favored 'Foundation' selected for use with RTCP.
   The connectivity checks for part 2 of the candidate component are
   shown in L (43-48) and R (49-54).  Once this has succeeded, User
   Agent L as the controlling client sends another connectivity check
   for the candidate pair.  This time the 'USE-CANDIDATE' attribute
   is again specified to signal the chosen candidate for component 2.

o  The candidates have now been fully verified (and selected), and as
   they are the highest priority, an updated offer (61-62) is now
   sent from the offerer (client L) to the answerer (client R)
   representing the new active candidates.  The new offer would look
   as follows:

```
v=0
o=test 2890844526 2890842808 IN IP4 $L-PRIV-1
c=IN IP4 $L-NAT-PUB-1.address
t=0 0
a=ice-pwd:$LPASS
a=ice-ufrag:$LUNAME
m=audio $L-NAT-PUB-1.port RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=rtcp:$L-NAT-PUB-2.port
a=candidate:$L2 1 UDP 2203948363 $L-NAT-PUB-1.address $L-NAT-PUB-1.port
            typ srflx raddr $L-PRIV-1.address rport $L-PRIV-1.port
a=candidate:$L2 2 UDP 2172635342 $L-NAT-PUB-2.address $L-NAT-PUB-2.port
            typ srflx raddr $L-PRIV-1.address rport $L-PRIV-2.port
```

                   Figure 16: ICE SDP Updated Offer

   o  The resulting answer (63-64) for R would look as follows:

```
v=0
o=test 3890844516 3890842804 IN IP4 $R-PRIV-1
c=IN IP4 $R-PRIV-1.address
t=0 0
a=ice-pwd:$RPASS
a=ice-ufrag:$RUNAME
m=audio $R-PRIV-1.port RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=rtcp:$R-PRIV-2.port
a=candidate:$L2 1 UDP 2984756463 $R-NAT-PUB-1.address $R-NAT-PUB-1.port
            typ srflx raddr $R-PRIV-1.address rport $R-PRIV-1.port
a=candidate:$L2 2 UDP 2605968473 $R-NAT-PUB-2.address $R-NAT-PUB-2.port
            typ srflx raddr $R-PRIV-1.address rport $R-PRIV-2.port
```

                   Figure 17: ICE SDP Updated Answer

5.2.2.  Address/Port-Dependent NAT

5.2.2.1.  STUN Failure

   This section highlights that although using STUN techniques is the
   preferred mechanism for traversal of NAT, it does not solve every
   case.  The use of basic STUN on its own will not guarantee traversal
   through every NAT type, hence the recommendation that ICE is the
   preferred option.

```
    Client      ADDRESS/PORT-Dependent    STUN              [..]
                       NAT                Server
      |                 |                   |                 |
      |(1) BIND Req     |                   |                 |
      |Src=L-PRIV-1     |                   |                 |
      |Dest=STUN-PUB    |                   |                 |
      |---------------->|                   |                 |
      |                 |(2) BIND Req       |                 |
      |                 |Src=NAT-PUB-1      |                 |
      |                 |Dest=STUN-PUB      |                 |
      |                 |---------------->  |                 |
      |                 |                   |                 |
      |                 |(3) BIND Resp      |                 |
      |                 |<----------------  |                 |
      |                 |Src=STUN-PUB       |                 |
      |                 |Dest=NAT-PUB-1     |                 |
      |                 |Map=NAT-PUB-1      |                 |
      |                 |                   |                 |
      |(4) BIND Resp    |                   |                 |
      |<----------------|                   |                 |
      |Src=STUN-PUB     |                   |                 |
      |Dest=L-PRIV-1    |                   |                 |
      |Map=NAT-PUB-1    |                   |                 |
      |                 |                   |                 |
      |(5)SIP INVITE    |                   |                 |
      |------------------------------------------------------->|
      |                 |                   |                 |
      |                 |                   |(6)SIP 200 OK    |
      |                 |<------------------------------------|
      |                 |                   |                 |
      |(7)SIP 200 OK    |                   |                 |
      |<----------------|                   |                 |
      |                 |                   |                 |
      |=======================================================|
      |>>>>>>>>>>>>>>Outgoing Media sent from L-PRIV-1>>>>>>>>>|
      |=======================================================|
      |                 |                   |                 |
      |                 x=====================================|
      |                 xIncoming Media sent to L-PRIV-1<<<<<<|
      |                 x=====================================|
      |                 |                   |                 |
      |(8)SIP ACK       |                   |                 |
      |---------------->|                   |                 |
      |                 |(9) SIP ACK        |                 |
      |                 |------------------------------------>|
      |                 |                   |                 |
```
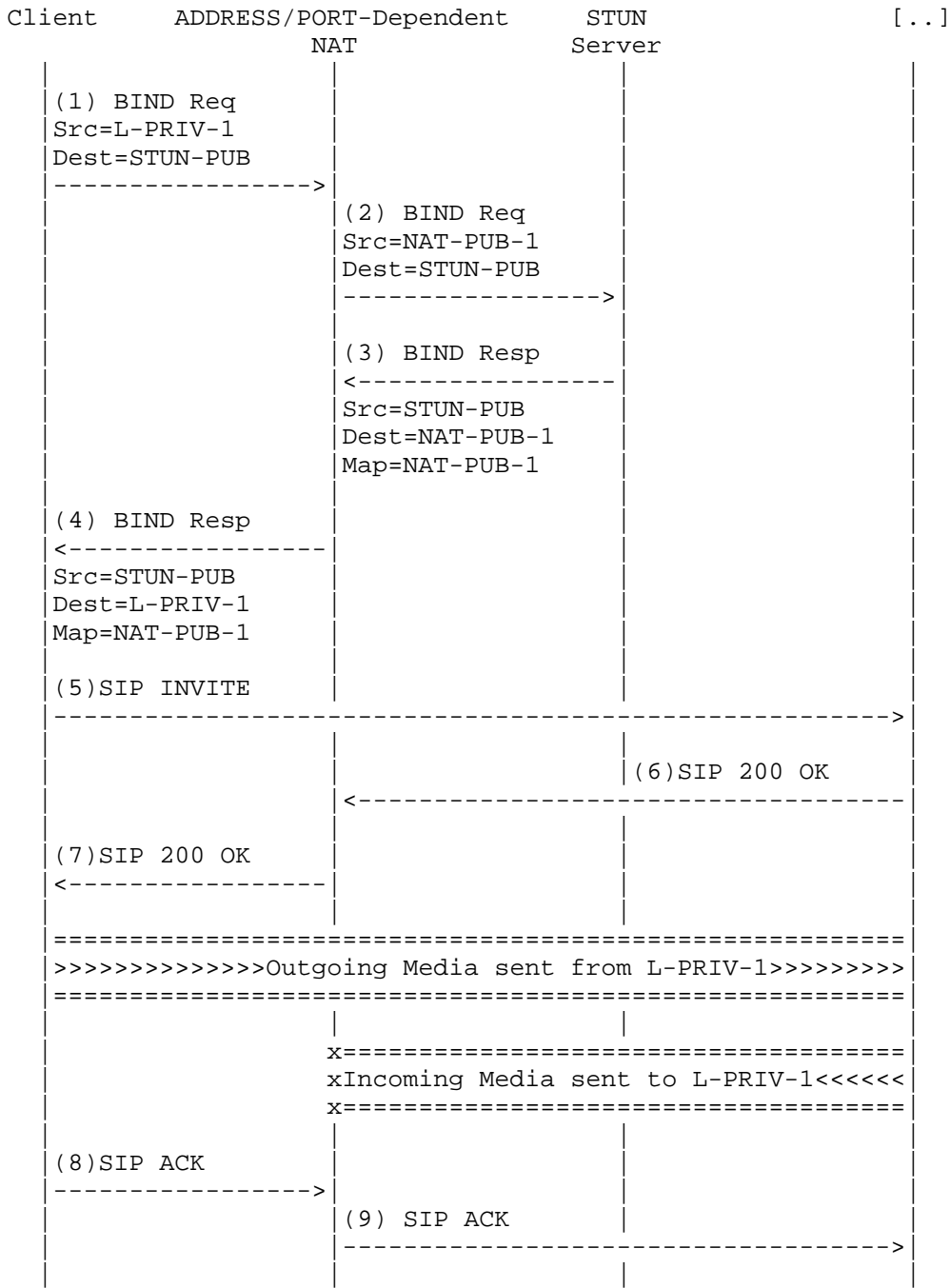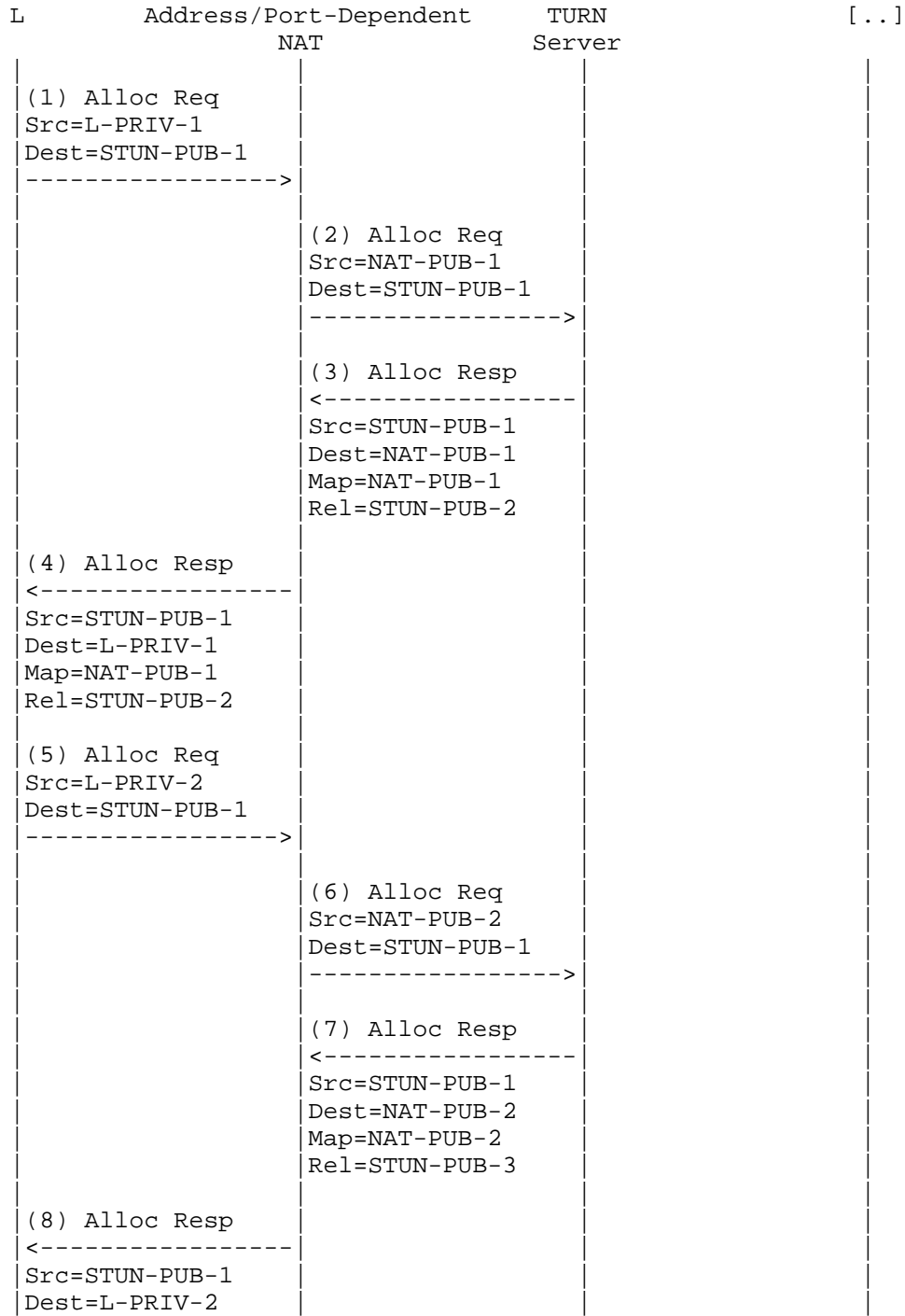
          Figure 18: Address/Port-Dependent NAT with STUN - Failure

The example in Figure 18 is conveyed in the context of a client
behind the Address/Port-Dependent NAT initiating a call.  It should
be noted that the same problem applies when a client receives a SIP
invitation and is behind a Address/Port-Dependent NAT.

o  In Figure 18, the client behind the NAT obtains a server reflexive
   representation using standard STUN mechanisms (1)-(4) that have
   been used in previous examples in this document (e.g.,
   Section 5.2.1.1.1).

o  The external mapped address (server reflexive) obtained is also
   used in the outgoing SDP contained in the SIP INVITE request (5).

o  In this example, the client is still able to send media to the
   external client.  The problem occurs when the client outside the
   NAT tries to use the reflexive address supplied in the outgoing
   INVITE request to traverse media back through the Address/
   Port-Dependent NAT.

o  A Address/Port-Dependent NAT has differing rules from the
   Endpoint-Independent type of NAT (as defined in RFC 4787
   [RFC4787]).  For any internal IP address and port combination,
   data sent to a different external destination does not provide the
   same public mapping at the NAT.  In Figure 18, the STUN query
   produced a valid external mapping for receiving media.  This
   mapping, however, can only be used in the context of the original
   STUN request that was sent to the STUN server.  Any packets that
   attempt to use the mapped address and that do not originate from
   the STUN server IP address and optionally port will be dropped at
   the NAT.  Figure 18 shows the media being dropped at the NAT after
   (7) and before (8).  This then leads to one-way audio.

5.2.2.2.  TURN Solution

   As identified in Section 5.2.2.1, STUN provides a useful tool for the
   traversal of the majority of NATs but fails with Address/
   Port-Dependent NAT.  The TURN extensions [RFC5766] address this
   scenario.  TURN extends STUN to allow a client to request a relayed
   address at the TURN server rather than a reflexive representation.
   This then introduces a media relay in the path for NAT traversal (as
   described in Section 4.2.3.2).  The following example explains how
   TURN solves the previous failure when using STUN to traverse a
   Address/Port-Dependent NAT.  It should be noted that TURN works most
   effectively when used in conjunction with ICE.  Using TURN on its own
   results in all media being relayed through a TURN server; this is not
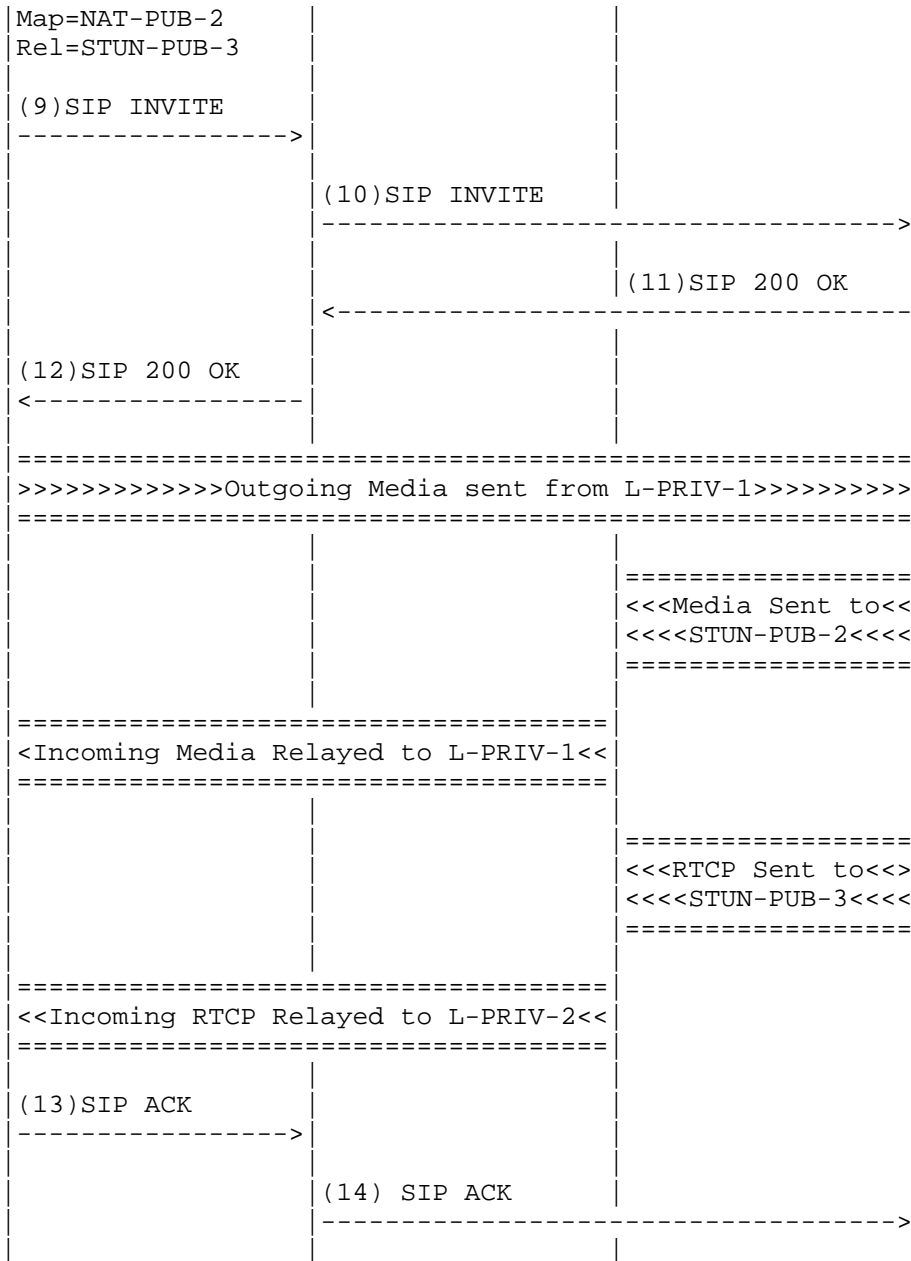   efficient.

```
         L          Address/Port-Dependent     TURN              [..]
                            NAT                 Server
         |                   |                   |                   |
         |(1) Alloc Req      |                   |                   |
         |Src=L-PRIV-1       |                   |                   |
         |Dest=STUN-PUB-1    |                   |                   |
         |----------------->|                   |                   |
         |                   |                   |                   |
         |                   |(2) Alloc Req      |                   |
         |                   |Src=NAT-PUB-1      |                   |
         |                   |Dest=STUN-PUB-1    |                   |
         |                   |---------------->|                   |
         |                   |                   |                   |
         |                   |(3) Alloc Resp     |                   |
         |                   |<----------------|                   |
         |                   |Src=STUN-PUB-1     |                   |
         |                   |Dest=NAT-PUB-1     |                   |
         |                   |Map=NAT-PUB-1      |                   |
         |                   |Rel=STUN-PUB-2     |                   |
         |                   |                   |                   |
         |(4) Alloc Resp     |                   |                   |
         |<----------------|                   |                   |
         |Src=STUN-PUB-1     |                   |                   |
         |Dest=L-PRIV-1      |                   |                   |
         |Map=NAT-PUB-1      |                   |                   |
         |Rel=STUN-PUB-2     |                   |                   |
         |                   |                   |                   |
         |(5) Alloc Req      |                   |                   |
         |Src=L-PRIV-2       |                   |                   |
         |Dest=STUN-PUB-1    |                   |                   |
         |----------------->|                   |                   |
         |                   |                   |                   |
         |                   |(6) Alloc Req      |                   |
         |                   |Src=NAT-PUB-2      |                   |
         |                   |Dest=STUN-PUB-1    |                   |
         |                   |----------------->|                   |
         |                   |                   |                   |
         |                   |(7) Alloc Resp     |                   |
         |                   |<----------------|                   |
         |                   |Src=STUN-PUB-1     |                   |
         |                   |Dest=NAT-PUB-2     |                   |
         |                   |Map=NAT-PUB-2      |                   |
         |                   |Rel=STUN-PUB-3     |                   |
         |                   |                   |                   |
         |(8) Alloc Resp     |                   |                   |
         |<----------------|                   |                   |
         |Src=STUN-PUB-1     |                   |                   |
         |Dest=L-PRIV-2      |                   |                   |
```

```
|Map=NAT-PUB-2     |                             |              |
|Rel=STUN-PUB-3    |                             |              |
|                  |                             |              |
|(9)SIP INVITE     |                             |              |
|---------------->|                             |              |
|                  |                             |              |
|                  |(10)SIP INVITE               |              |
|                  |---------------------------------->|        |
|                  |                             |              |
|                  |                             |(11)SIP 200 OK|
|                  |<----------------------------------|        |
|                  |                             |              |
|(12)SIP 200 OK    |                             |              |
|<----------------|                             |              |
|                  |                             |              |
|==================================================================|
|>>>>>>>>>>>>>Outgoing Media sent from L-PRIV-1>>>>>>>>>>|
|==================================================================|
|                  |                             |              |
|                  |                             |================|
|                  |                             |<<<Media Sent to<<|
|                  |                             |<<<<STUN-PUB-2<<<<|
|                  |                             |================|
|                  |                             |              |
|==================================|              |              |
|<Incoming Media Relayed to L-PRIV-1<<|            |              |
|==================================|              |              |
|                  |                             |              |
|                  |                             |================|
|                  |                             |<<<RTCP Sent to<<>|
|                  |                             |<<<<STUN-PUB-3<<<<|
|                  |                             |================|
|                  |                             |              |
|==================================|              |              |
|<<Incoming RTCP Relayed to L-PRIV-2<<|           |              |
|==================================|              |              |
|                  |                             |              |
|(13)SIP ACK       |                             |              |
|---------------->|                             |              |
|                  |                             |              |
|                  |(14) SIP ACK                 |              |
|                  |---------------------------------->|        |
|                  |                             |              |
```

Figure 19: Address/Port-Dependent NAT with TURN - Success

o  In this example, client L issues a TURN allocate request (1) to
   obtained a relay address at the STUN server.  The request
   traverses through the Address/Port-Dependent NAT and reaches the
   STUN server (2).  The STUN server generates an Allocate response
   (3) that contains both a server reflexive address (Map=NAT-PUB-1)
   of the client and also a relayed address (Rel=STUN-PUB-2).  The
   relayed address maps to an address mapping on the STUN server that
   is bound to the public pinhole that has been opened on the NAT by
   the Allocate request.  This results in any traffic sent to the
   TURN server relayed address (Rel=STUN-PUB-2) being forwarded to
   the external representation of the pinhole created by the Allocate
   request (NAT-PUB-1).

o  The TURN derived address (STUN-PUB-2) arrives back at the
   originating client (4) in an Allocate response.  This address can
   then be used in the SDP for the outgoing SIP INVITE request as
   shown in the following example (note that the example also
   includes client L obtaining a second relay address for use in the
   RTCP attribute (5-8)):

   v=0
   o=test 2890844342 2890842164 IN IP4 $L-PRIV-1
   c=IN IP4 $STUN-PUB-2.address
   t=0 0
   m=audio $STUN-PUB-2.port RTP/AVP 0
   a=rtcp:$STUN-PUB-3.port

o  On receiving the INVITE request, the User Agent Server (UAS) is
   able to stream media and RTCP to the relay address (STUN-PUB-2 and
   STUN-PUB-3) at the STUN server.  As shown in Figure 19 (between
   messages (12) and (13), the media from the UAS is directed to the
   relayed address at the STUN server.  The STUN server then forwards
   the traffic to the open pinholes in the Address/Port-Dependent NAT
   (NAT-PUB-1 and NAT-PUB-2).  The media traffic is then able to
   traverse the Address/Port-Dependent NAT and arrives back at client
   L.

o  TURN on its own will work for Address/Port-Dependent and other
   types of NAT mentioned in this specification but should only be
   used as a last resort.  The relaying of media through an external
   entity is not an efficient mechanism for NAT traversal and comes
   at a high processing cost.

5.2.2.3.  ICE Solution

   The previous two examples have highlighted the problem with using
   core STUN for all forms of NAT traversal and a solution using TURN
   for the Address/Port-Dependent NAT case.  The RECOMMENDED mechanism

for traversing all varieties of NAT is using ICE, as detailed in
Section 4.2.3.3.  ICE makes use of core STUN, TURN and any other
mechanism (e.g., NAT-PMP[NAT-PMP], UPnP IGD[UPnP-IGD]) to provide a
list of prioritized addresses that can be used for media traffic.
Detailed examples of ICE can be found in Section 5.2.1.2.1.  These
examples are associated with an Endpoint-Independent type NAT but can
be applied to any NAT type variation, including Address/
Port-Dependent type NAT.  The ICE procedures carried out are the
same.  For a list of candidate addresses, a client will choose where
to send media dependent on the results of the STUN connectivity
checks and associated priority (highest priority wins).  It should be
noted that the inclusion of a NAT displaying Address/Port-Dependent
properties does not automatically result in relayed media.  In fact,
ICE processing will avoid use of media relay with the exception of
two clients that both happen to be behind a NAT using Address/
Port-Dependent characteristics.  The connectivity checks and
associated selection algorithm enable traversal in this case.
Figure 20 and the following description provide a guide as to how
this is achieved using the ICE connectivity checks.  This is an
abbreviated example that assumes successful SIP offer/answer exchange
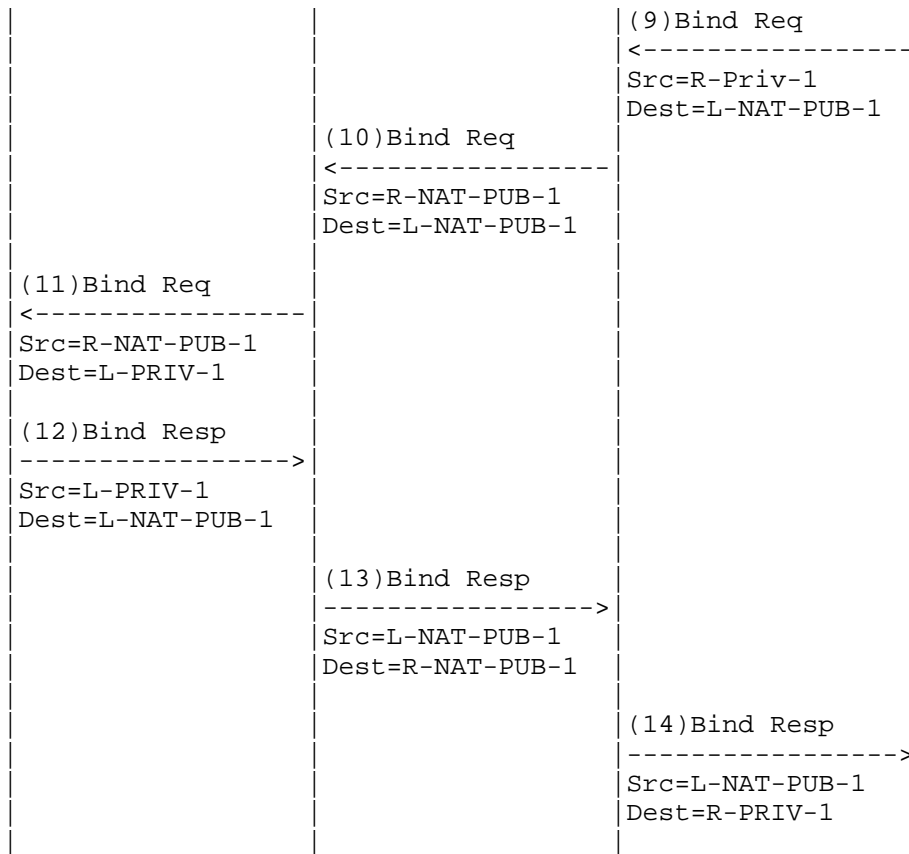and illustrates the connectivity check flow.

```
        L       Address/Port-Dependent  Endpoint-Independent      R
                         L-NAT                   R-NAT
        |=========================================================|
        |                 SIP OFFER/ANSWER EXCHANGE                |
        |=========================================================|
        |                 |                      |                 |
        |                 |                      |(1)Bind Req      |
        |                 |                      |<----------------|
        |                 |                      |Src=R=PRIV-1     |
        |                 |                      |Dest=L-NAT-PUB-1 |
        |                 |                      |                 |
        |                 |(2)Bind Req           |                 |
        |                x<----------------|     |                 |
        |                 |Src=R-NAT-PUB-1       |                 |
        |                 |Dest=L-NAT-PUB-1      |                 |
        |                 |                      |                 |
        |(3)Bind Req      |                      |                 |
        |---------------->|                      |                 |
        |Src=L-PRIV-1     |                      |                 |
        |Dest=R-NAT-PUB-1 |                      |                 |
        |                 |                      |                 |
        |                 |(4)Bind Req           |                 |
        |                 |---------------->|     |                 |
        |                 |Src=L-NAT-PUB-1       |                 |
        |                 |Dest=R-NAT-PUB-1      |                 |
        |                 |                      |                 |
        |                 |                      |(5)Bind Req      |
        |                 |                      |---------------->|
        |                 |                      |Src=L-NAT-PUB-1  |
        |                 |                      |Dest=R-PRIV-1    |
        |                 |                      |                 |
        |                 |                      |(6)Bind Resp     |
        |                 |                      |<----------------|
        |                 |                      |Src=R-PRIV-1     |
        |                 |                      |Dest=L-NAT-PUB-1 |
        |                 |                      |                 |
        |                 |(7)Bind Resp          |                 |
        |                 |<----------------|     |                 |
        |                 |Src=R-NAT-PUB-1       |                 |
        |                 |Dest=L-NAT-PUB-1      |                 |
        |                 |                      |                 |
        |(8)Bind Resp     |                      |                 |
        |<----------------|                      |                 |
        |Src=R-NAT-PUB-1  |                      |                 |
        |Dest=L-PRIV-1    |                      |                 |
        |                 |                      |                 |
```

```
    |                  |                  |(9)Bind Req       |
    |                  |                  |<-----------------|
    |                  |                  |Src=R-Priv-1      |
    |                  |                  |Dest=L-NAT-PUB-1  |
    |                  |(10)Bind Req      |                  |
    |                  |<-----------------|                  |
    |                  |Src=R-NAT-PUB-1   |                  |
    |                  |Dest=L-NAT-PUB-1  |                  |
    |                  |                  |                  |
    |(11)Bind Req      |                  |                  |
    |<-----------------|                  |                  |
    |Src=R-NAT-PUB-1   |                  |                  |
    |Dest=L-PRIV-1     |                  |                  |
    |                  |                  |                  |
    |(12)Bind Resp     |                  |                  |
    |----------------->|                  |                  |
    |Src=L-PRIV-1      |                  |                  |
    |Dest=L-NAT-PUB-1  |                  |                  |
    |                  |                  |                  |
    |                  |(13)Bind Resp     |                  |
    |                  |----------------->|                  |
    |                  |Src=L-NAT-PUB-1   |                  |
    |                  |Dest=R-NAT-PUB-1  |                  |
    |                  |                  |                  |
    |                  |                  |(14)Bind Resp     |
    |                  |                  |----------------->|
    |                  |                  |Src=L-NAT-PUB-1   |
    |                  |                  |Dest=R-PRIV-1     |
    |                  |                  |                  |
```

       Figure 20: Single Address/Port-Dependent NAT - Success

   In this abbreviated example, client R has already received a SIP
   INVITE request and is starting its connectivity checks with client L.
    Client R generates a connectivity check (1) and sends to client L's
   information as presented in the SDP offer.  The request arrives at
   client L's Address/Port-Dependent NAT and fails to traverse as there
   is no NAT binding.  This would then move the connectivity check to a
   failed state.  In the meantime, client L has received the SDP answer
   in the SIP request and will also commence connectivity checks.  A
   check is dispatched (3) to client R.  The check is able to traverse
   the NAT due to the association set up in the previously failed check
   (1).  The full Bind request/response is shown in steps (3)-(8).  As
   part of a candidate pair, client R will now successfully be able to
   complete the checks, as illustrated in steps (9)-(14).  The result is
   a successful pair of candidates that can be used without the need to
   relay any media.

In conclusion, the only time media needs to be relayed is a result of
clients both behind Address/Port-Dependent NATs.  As you can see from
the example in this section, neither side would be able to complete
connectivity checks with the exception of the Relayed candidates.

6.  IPv4-IPv6 Transition

This section describes how IPv6-only SIP User Agents can communicate
with IPv4-only SIP User Agents.  While the techniques discussed in
this document primarily contain examples of traversing NATs to allow
communications between hosts in private and public networks, they are
by no means limited to such scenarios.  The same NAT traversal
techniques can also be used to establish communication in a
heterogeneous network environment -- e.g., communication between an
IPv4 host and an IPv6 host.

6.1.  IPv4-IPv6 Transition for SIP Signaling

IPv4-IPv6 translations at the SIP level usually take place at dual-
stack proxies that have both IPv4 and IPv6 DNS entries.  Since these
translations do not involve NATs that are placed in the middle of two
SIP entities, they fall outside the scope of this document.  A
detailed description of this type of translation can be found in
[RFC6157].

7.  Security Considerations

There are no security considerations beyond the ones inherited by
reference.

8.  Acknowledgments

The authors would like to thank the members of the IETF SIPPING WG
for their comments and suggestions.  Expert review and detailed
contribution including text was provided by Dan Wing, who was
supportive throughout.

Detailed comments were provided by Vijay Gurbani, Kaiduan Xie, Remi
Denis-Courmont, Hadriel Kaplan, Phillip Matthews, Spencer Dawkins,
and Hans Persson.

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.

   [RFC3263]       Rosenberg, J. and H. Schulzrinne, "Session Initiation
                   Protocol (SIP): Locating SIP Servers", RFC 3263,
                   June 2002.

   [RFC3264]       Rosenberg, J. and H. Schulzrinne, "An Offer/Answer
                   Model with Session Description Protocol (SDP)",
                   RFC 3264, June 2002.

   [RFC3327]       Willis, D. and B. Hoeneisen, "Session Initiation
                   Protocol (SIP) Extension Header Field for Registering
                   Non-Adjacent Contacts", RFC 3327, December 2002.

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

   [RFC3581]       Rosenberg, J. and H. Schulzrinne, "An Extension to the
                   Session Initiation Protocol (SIP) for Symmetric
                   Response Routing", RFC 3581, August 2003.

   [RFC3605]       Huitema, C., "Real Time Control Protocol (RTCP)
                   attribute in Session Description Protocol (SDP)",
                   RFC 3605, October 2003.

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

   [RFC4787]       Audet, F. and C. Jennings, "Network Address
                   Translation (NAT) Behavioral Requirements for Unicast
                   UDP", BCP 127, RFC 4787, January 2007.

   [RFC4961]       Wing, D., "Symmetric RTP / RTP Control Protocol
                   (RTCP)", BCP 131, RFC 4961, July 2007.

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

   [RFC5389]        Rosenberg, J., Mahy, R., Matthews, P., and D. Wing,
                    "Session Traversal Utilities for NAT (STUN)",
                    RFC 5389, October 2008.

   [RFC5626]        Jennings, C., Mahy, R., and F. Audet, "Managing
                    Client-Initiated Connections in the Session Initiation
                    Protocol (SIP)", RFC 5626, October 2009.

   [RFC5761]        Perkins, C. and M. Westerlund, "Multiplexing RTP Data
                    and Control Packets on a Single Port", RFC 5761,
                    April 2010.

   [RFC5766]        Mahy, R., Matthews, P., and J. Rosenberg, "Traversal
                    Using Relays around NAT (TURN): Relay Extensions to
                    Session Traversal Utilities for NAT (STUN)", RFC 5766,
                    April 2010.

   [RFC5923]        Gurbani, V., Mahy, R., and B. Tate, "Connection Reuse
                    in the Session Initiation Protocol (SIP)", RFC 5923,
                    June 2010.

## 9.2.  Informative References

   [MIDDLEBOXES]    Stucker, B. and H. Tschofenig, "Analysis of Middlebox
                    Interactions for Signaling Protocol Communication
                    along the Media Path", Work in Progress, July 2010.

   [NAT-PMP]        Cheshire, S., "NAT Port Mapping Protocol (NAT-PMP)",
                    Work in Progress, April 2008.

   [RFC2026]        Bradner, S., "The Internet Standards Process --
                    Revision 3", BCP 9, RFC 2026, October 1996.

   [RFC3424]        Daigle, L. and IAB, "IAB Considerations for UNilateral
                    Self-Address Fixing (UNSAF) Across Network Address
                    Translation", RFC 3424, November 2002.

   [RFC5780]        MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery
                    Using Session Traversal Utilities for NAT (STUN)",
                    RFC 5780, May 2010.

      [RFC5853]       Hautakorpi, J., Camarillo, G., Penfield, R.,
                      Hawrylyshen, A., and M. Bhatia, "Requirements from
                      Session Initiation Protocol (SIP) Session Border
                      Control (SBC) Deployments", RFC 5853, April 2010.

      [RFC6157]       Camarillo, G., El Malki, K., and V. Gurbani, "IPv6
                      Transition in the Session Initiation Protocol (SIP)",
                      RFC 6157, April 2011.

      [UPnP-IGD]      UPnP Forum, "Universal Plug and Play Internet Gateway
                      Device v1.0", 2000,
                      <http://www.upnp.org/specs/gw/igd1/>.

Authors' Addresses

   Chris Boulton
   NS-Technologies

   EMail: chris@ns-technologies.com


   Jonathan Rosenberg
   Skype

   EMail: jdrosen@jdrosen.net


   Gonzalo Camarillo
   Ericsson
   Hirsalantie 11
   Jorvas  02420
   Finland

   EMail: Gonzalo.Camarillo@ericsson.com


   Francois Audet
   Skype

   EMail: francois.audet@skype.net