SIPPING
Internet-Draft
Expires: December 29, 2003

J. Rosenberg
dynamicsoft
June 30, 2003

# Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for the Session Initiation Protocol (SIP)

draft-rosenberg-sipping-ice-01

Abstract

This document describes a methodology for Network Address Translator (NAT) traversal for the Session Initiation Protocol (SIP). This methodology is called Interactive Connectivity Establishment (ICE). ICE is not a new protocol, but rather makes use of existing protocols, such as Simple Traversal of UDP Through NAT (STUN), Traversal Using Relay NAT (TURN) and even Real Specific IP (RSIP). ICE works through the mutual cooperation of both endpoints in a SIP dialog.

Table of Contents

## 1. Introduction

The subject of NAT traversal for SIP has received a profound amount of attention. SIP extensions have been defined for routing responses [11] through NAT, and for routing requests from a public network to a private one through persistent connections [12].

However, far more troubling is the traversal of SIP's media sessions through NAT. Numerous solutions have been proposed for that. These include Application Layer Gateways (ALGs), the Middlebox Control Protocol [2], Simple Traversal of UDP through NAT (STUN) [13], Traversal Using Relay NAT [14], Realm Specific IP [3][4], Topology Insensitive Service Traversal (TIST) [15], symmetric RTP [16], along with protocol extensions needed to make them work, such as [17]. The sum total is so complex, that an extensive scenarios document [18] has been written. The latter outlines various network situations, and analyzes the various solutions in each. Unsurprisingly, each situation has a specific ideal solution.

However, the result is a system which is incredibly complex and very brittle. What is needed is a single solution which is flexible enough to work well in all situations.

Our proposal for such a solution is called Interactive Connectivity Establishment, or ICE. ICE makes use of many of the protocols above, but uses them in a specific methodology which avoids many of the pitfalls of using any one alone. ICE is not a new protocol, and does not require extensions from STUN, TURN or RSIP. However, it does require some additional SDP attributes, which are discussed below.

## 2. Overview of ICE

ICE makes the fundamental assumption that clients exist in a network of segmented connectivity. This segmentation is the result of a number of addressing realms in which a client can simultaneously be connected. We use "realms" here in the broadest sense. A realm is defined purely by connectivity. Two clients are in the same realm if, when they exchange the addresses each has in that realm, they are able to send packets to each other. This includes IPv6 and IPv4 realms, which actually use different address spaces, in addition to private networks connected to the public Internet through NAT.

The key assumption in ICE is that a client cannot know, apriori, whether the peer it wishes to communicate with is connected to one or all of the address realms it is in. Therefore, in order to communicate, it has to try them all, and choose the best one that works.

Before a UA makes a call, it obtains as many IP address and port combinations in as many address realms as it can. These adresses all represent potential points at which the UA will receive a specific media stream. Any protocol that provides a client with an IP address and port on which it can receive traffic can be used. These include STUN, TURN, RSIP, and even a VPN. The client also uses any local interface addresses. A dual-stack v4/v6 client will obtain both a v6 and a v4 address/port. The only requirement is that, across all of these addresses, the client can be certain that at least one of them will work for any peer. This is easily guaranteed by using TURN, RSIP, MIDCOM or a VPN from a server on the public Internet to obtain one of the addresses.

The UAC then makes a STUN server available on each of the address/ port combinations it has obtained. This STUN server is running locally, on the UA.

All of these addresses are placed into the SDP offer [5] sent by the UAC. Each of them is represented by a separate SDP attribute, called "alt". They are ordered in terms of preference. Local IPv6 addresses always have the highest preference, followed by local IPv4 addresses, followed by STUN-allocated addresses, followed last by addresses allocated through protocols using relays, such as TURN and VPN. The alt attribute is also used to convey the STUN username and password which are required to gain access to the STUN server on each address/ port combination.

This offer is sent to the called party. ICE also assumes that SIP itself can provide global connectivity across address realms. Indeed, the point of the SIP URI is to act as a globally useful identifier

for reaching a user wherever they are.

Once the offer arrives at the UAS, it sends STUN requests to each alternate address/port in the SDP offer, similar to the intra-realm STUN mechanism [21] proposed previously. These STUN requests include the username and password obtained from the SDP. None of the flags are used. The STUN requests serve two purposes. The first is to check for connectivity. If a response is received, the UAS knows that it can reach the UAC at that address. The second purpose is to obtain more addresses at which the UAS can be contacted. If there were NATs between the UAS and UAC, the UAS may discover another address through the STUN responses. In its answer, the UAS includes all addresses that it can unilaterally determine (just as the UAC did), in addition to any that were discovered using the STUN messages to the UAC.

When the answer arrives at the UAC, it performs a similar operation. Using STUN, it checks connectivity to each of the addresses in the answer. Through the STUN responses, it may learn of additional addresses that it can use to receive media. It can therefore generate a re-INVITE or UPDATE [6] request to pass this address to the callee. Generally, at the end of the first exchange, both sides will have discovered one of more addresses which they are capable of successfully sending to. Each side uses the most preferred address amongst the ones which worked.

## 3. Terminology

Several new terms are introduced in this specification:

Transport Address: The combination of an IP address and port.

Local Transport Address: A local transport address is transport address that has been allocated from the operating system on the host. This includes transport addresses obtained through VPNs, and also transport addresses obtained through RSIP (which lives at the operating system level). Transport addresses are typically obtained by binding to an interface.

Derived Transport Address: A derived transport address is a transport address which is associated with, but different from, a local transport address. The derived transport address is associated with the local transport address in that packets sent to the derived transport address are received on the socket bound to that local transport address. Derived addresses are obtained using protocols like STUN and TURN, and more generally, any UNSAF protocol [8].

Peer Derived Transport Address: A peer derived transport address is a derived transport address learned from a STUN server advertised by a peer in a media session.

TURN Derived Transport Address: A derived transport address obtained from a TURN server.

STUN Derived Transport Address: A derived transport address obtained from a STUN server that has been provisioned into the UA. This, by definition, excludes Peer Derived Transport Addresses.

Unilateral Allocations: Queries made to a network server which provides an UNSAF service.

## 4. Core ICE Algorithm

At its core, the ICE algorithm is an iterative process in which two cooperating entities, A and B, exchange addresses with each other in an attempt to connect. One side (say A) starts, collecting all of the addresses it can find. It sends those to B. B also collects all of the addresses it can find, including those obtained by sending address-fixing requests (such as STUN requests) to A itself. Those are passed to A. B also checks connectivity to the addresses provided by A. When A gets the set of addresses, it performs connectivity checks, and attempts to obtain further addresses based on the information sent by B. If A learns more addresses, it sends these to B, which checks connectivity to those addresses. This process iterates back and forth until both sides have obtain all the addresses which can be obtained. At least one address passed in each direction should work.

## 5. Detailed ICE Algorithm

This section describes the detailed processing needed for ICE.

### 5.1 Gathering Transport Addresses

The offerer begins the process by gathering transport addresses. There are two types of addresses it can gather - local transport addresses, and derived transport addresses. Local transport addresses are obtained by binding to an ephemeral port on an interface (physical or virtual) on the host. A multi-homed host SHOULD attempt to bind on all interfaces for all media streams it wishes to receive. For media streams carried using the Real Time Transport Protocol (RTP) [10], the offerer will need to bind to an ephemeral port for both RTP and RTCP.

The result will be a set of local transport addresses. The offerer may also have access to servers that provide unilateral self-address fixing (UNSAF) [8]. Examples of such protocols include STUN, TURN, and TEREDO [22]. For each of these protocols, the offerer may have access to a multiplicity of servers. For example, a user connected to a natted cable access network might have access to a STUN server in the private cable network and in the public Internet. For each local transport address, the offerer SHOULD address-fix against every server for each protocol it supports. The result of this will be a set of derived transport addresses, with each derived address associated with the local transport address it is derived from.

ICE works better the more options exist for connectivity. However, in order to communicate with the peer, at least one of the offered addresses has to be guaranteed to work with any peer that might be called. This generally requires that one of the derived addresses be obtained from a relay service (such as TURN or TEREDO) that exist within the public Internet.

### 5.2 Enabling STUN on Each Transport Address

Once the offerer has obtained a set of transport addresses, it starts a STUN server on each local transport address (including ones used for RTCP). This, by definition, means that the STUN service will be reached for requests sent to the derived addresses.

However, the client does not need to provide STUN service on any other IP address or port, unlike the normal STUN usage as described in [13]. The need to run the service on multiple ports is to support the change flags. However, those flags are not needed with ICE, and the server SHOULD reject any requests with these flags set.

Furthermore, there is no need to support TLS or to be prepared to receive SharedSecret request messages. Those messages are used to obtain shared secrets to be used with BindingRequests. However, with ICE, usernames and passwords are exchanged in SIP itself.

It is important to note that the transport address being used by the STUN server will also need to support the media stream which is to be sent to that transport address. This will require the offerer to disambiguate STUN messages from messages for the underlying media stream protocol. In the case of RTP/RTCP, this disambiguation is easy. RTP and RTCP packets start with the bits 0b10 (v=2). The first two bits in STUN are always 0b00. Disambiguating STUN with other media stream protocols may be more complicated. However, it is guaranteed to always be possible by selecting an appropriately random username (see below).

The need to run STUN on the same transport address as the media stream represents the "ugliest" piece of ICE. However, it is an essential part of the story. By sending STUN requests to the very same place media is sent, any bindings learned through STUN will be useful even when communicating through symmetric NATs. This results in a substantial increase in the scope of applicability of STUN, in terms of cases where it can provide connectivity. In that sense, the usage of STUN here is radically different than the usage models outlined in [13], where STUN is generally useless for dealing with symmetric NAT.

For each local transport address where a STUN server is running, the client MUST choose a username and password. The username MUST be globally unique, so that no other host will select a username with the same value. This username and password will be passed to the answerer in the SDP. They are used by the answerer to authenticate the STUN requests to the server.

The global uniqueness requirement stems from the lack of uniquenes afforded by IP addresses. Consider user agents A, B, and C. A and B are within private enterprise 1, which is using 10.0.0.0/8. C is within private enterprise 2, which is also using 10.0.0.0/8. As it turns out, B and C both have IP address 10.0.1.1. A makes a call to C. C, in its answer, provides A with its transport addresses. In this case, thats 10.0.1.1:8866 and 8877. As it turns out, B is on a call at that same time, and is also using 10.0.1.1:8866 and 8877. This means that B has a STUN server running on those ports, just as C does. A will send a STUN request to 10.0.1.1:8866 and 8877. However, these do not go to C as expected. Instead, they go to B. If B just replied to them, A would believe it has connectivity to C, when in fact it has connectivity to a completely different user, B. To fix this, the STUN username takes on the role of a unique identifier. C

provides A with a unique username. A uses this username in its STUN query to 10.0.1.1:8866. This STUN query arrives at B. However, the username is unknown to B, and so the request is rejected. A treats the rejected STUN request as if there were no connectivity to C (which is actually true). Therefore, the error is avoided.

Once the STUN server is started, it MUST run until the first media packet arrives on that address. Once that occurs, the agent MAY terminate the server. It is still possible that a late or lost STUN message will show up, but these will generally fail any media stream validity checks and be discarded (STUN packets always fail the RTP validity checks).

While the server is running, it MUST act as a normal STUN server, but MUST only accept STUN requests from clients that authenticate using the username and password handed out for the dialog.

### 5.3 Prioritizing the Transport Addresses

With the STUN servers starting, the next step is to prioritize the transport addresses. This priority reflects the desire that the UA has to receive media on that address, and is assigned as a value from 0 to 1 (1 being most preferred). Although any prioritization is possible, it is RECOMMENDED that the prioritization be based on the number of intermediaries that will be traversed. The fewer intermediaries, the higher the priority. Furthermore, it is RECOMMENDED that, given an equal number of intermediaries, an IPv6 address receive higher priority than an IPv4 address. As a result of this, local IPv6 transport addresses obtained from physical interfaces have highest priority. Next are local IPv4 transport addresses obtained from physical interfaces. Next are STUN derived transport addresses, followed by TURN, RSIP or TEREDO derived transport addresses. Last up are local transport addresses obtained from VPN interfaces.

### 5.4 Constructing the Offer

The next step is to construct the offer. For each media stream, the client chooses the transport address which has the highest probability of working with any arbitrary peer. Generally, this will be a transport address learned from a relay service on the public Internet, such as TURN. Frequently this is also the lowest priority transport address. This transport address is placed into the m and c lines of the offer. This is the address that will be used by a peer that doesn't understand ICE. Therefore, to maximize the ability to complete a call, the address which is most likely to succeed is used. In the case of RTP, the corresponding RTCP address would be placed into the rtcp attribute [17] if its not on the port one higher than

the RTP.

> OPEN ISSUE: There are cases where there is no clear "highest probability" address. The example intra-enterprise call shows such a case. The TURN relay returns two addresses, and the right one to use as a default depends on who you are calling. This may be unavoidable.

The client then encodes all of its available transport addresses (including the one that was just placed into the m and c lines) as a series of alt attributes (see Section 7. Each alt attribute conveys a single transport address (or, in the case of RTP, two transport addresses - one for RTP, and one for RTCP). Each will have a unique identifier. The priority for the transport address, as computed above, is included in the attribute as well.

The usage of the alt attribute implies that a STUN server is running on the transport addresses associated with that attribute. In the case of RTP, this means that STUN servers are running on both the RTP and RCP ports, independently of whether the RTCP port is equal to the RTP port plus one, or explicitly signaled using the second transport address. The alt attribute also contains the username and password to be used for sending STUN requests.

Once the offer is constructed, it is sent.

> The previous version of this specification used the ALT grouping semantic [20]. However, this revision uses a new alt attribute in order to improve backwards compatibility between ICE and non-ICE clients. These attributes are ignored by non-ICE clients, and the result is that media is sent to a single address and port - the one present in the m and c lines. By choosing those to be the TURN-allocated address, we maximize the likelihood of successful call completion even when communicating to a non-ICE client.

### 5.5 Answerer Processing - Connectivity Checks and Gathering

Once the answerer receives the offer, it does several things in parallel. First, it performs the same processing described in Section 5.1. Specifically, for each media stream in the offer, , the answerer allocates a set of local transport addresses and the full set of derived transport addresses.

Note that these addresses can be "pre-gathered" before the call is even received, so that a set is always "on-deck". This will avoid any increase in call setup times, at the expense of holding onto addresses which may not get used. Retaining these addresses may also

require refresh traffic that consumes network bandwidth.

While the unilateral derived addresses are being obtained, the answerer sends a STUN BindingRequest from each local transport address to each STUN server identified in an alt attribute in the offer. This BindingRequest MUST contain the USERNAME attribute, and the value of the USERNAME MUST equal the username from that alt attribute. Similarly, the BindingRequest MUST contain a PASSWORD attribute, and the value of the PASSWORD MUST equal the password from the alt attribute. The BindingRequest MUST contain a MESSAGE-INTEGRITY attribute, computed using the username and password from the alt attribute in the offer. The BindingRequest MUST NOT contain the CHANGE-REQUEST or RESPONSE-ADDRESS attribute.

It is RECOMMENDED that these STUN requests be sent in parallel. The answerer MAY alert the user during this time. Generally, if the user is a human (and not an automata), the STUN transactions will have completed before the call is answered.

If the STUN BindingRequest elicits a BindingResponse before the STUN transaction times out, the result is considered a success. For successful transactions, the answerer stores the offered transport address (which identifies both the STUN server and the place where media is sent), the local transport address from which the STUN request was sent, the id of that alternative from the offer, and the qvalue from that alternative. If the STUN transaction succeeds, the client knows for certain that the media can reach the destination as well. That is because the media traffic will be sent from the same transport address, to the same trasport address, as the STUN packet was.

When a client receives an offer, it MAY begin sending media immediately to the address and port in the m and c-lines. If that address and port was also encoded in an alt attribute, the client MUST send media from the same address and port used to send a STUN request to the peer. As the STUN transactions begin to complete, the client begins to learn about alternative addresses which have connectivity. If one of those alternatives has a higher priority than the one currently in use, that transport address MUST be used instead (along with its corresponding local address). Note that, between two transport addresses with the same q-value, a STUN derived address always has higher priority. Furthermore, once an agent sends media to a transport address with a specified priority, it MUST NOT, during the lifetime of the dialog, send media to a connected transport address with a lower priority.

This restriction allows an agent to free derived transport addresses once it knows that its peer has been able to connect to a transport

address with higher priority, or one of equal priority if it was peer derived. An agent can know that a peer was able to connect to a transport address based on the receipt of a STUN BindingRequest against that transport address. The username and password in the STUN BindingRequest can be used to determine which transport address the STUN request was generated against. Note that the transport address that the STUN request was received on does not say anything about which transport address the peer sent to, and so the username and password are used. Such an address SHOULD be freed no earlier than 3 seconds after receipt of the STUN request. This provides a window of time for the peer to cease using the address and switch to a better one.

This connectivity check makes an important assumption. It assumes that if a STUN request is able to get from A to B, the STUN response will get from B to A (packet losses aside). To our knowledge this is generally true, since it is one of the definining characteristics of the client-server protocols that NATs have been designed to pass. We need to make this assumption in order to free up resources and also eliminate additional ICE cycles.

The drawback of this restriction is that if connectivity should be lost during the dialog, the client cannot fall back to lower priority address. We believe that it is more important to free unneeded resources than to hold onto them in case of the unlikely event of a problem.

For those successful STUN transactions, the answerer compares the MAPPED-ADDRESS attribute in the response to the local transport address from which the STUN request was sent. If the two differ, the answerer considers the MAPPED-ADDRESS as another transport address that has been gathered for usage in this dialog. This transport address is referred to as a peer derived transport address. The priority of this transport address is set to the value of the qvalue of that alternative from the offer. For example, if the offerer provides a transport address obtained from a local interface, it would set the qvalue to 1.0. If the answerer sends a STUN request to the server and obtains a new transport address, that transport address is assigned a qvalue of 1.0. That qvalue will be used in comparison to other addresses gathered by the answerer.

If any STUN BindingRequest results in a BindingErrorResponse, the ERROR-CODE is examined. If it is 401, 430, 432 or 500, the client SHOULD retry the request, applying any appropriate fixes specified by the error code. In the case of 400, 431 and 600, the client MUST NOT retry. This case is treated identically to a timeout, so that it is equal to no connectivity at all.

## 5.6 Generating the Answer

At some point, the called party will decide to accept or reject the call. A rejection terminates ICE processing, of course. In the case of acceptance, the answer is constructed as follows.

At the time when the answer is to be sent, the answerer will have gathered some number of transport addresses. Some of these will be local transport addresses, some will be unilaterally derived addresses, and some will be stun derived from the peer in the dialog. Each of these will have a qvalue, based on either the rules in Section 5.1 or Section 5.5.

Constructing the answer proceeds identically to the way in which the offer is constructed (Section 5.4). The transport address with the highest probability of success is placed into the m and c lines. Furthermore, all of the alternatives are placed into an "alt" attribute. Each is assigned a unique ID. Those addresses which were peer-derived STUN addresses contain the "id" attribute identifying the address they were derived from. Each alternative contains its qvalue as computed above. Each alternative contains a username and password that must be used to contact the STUN server listening on that address. Each address SHOULD have a different username and password.

The answer is then sent.

## 5.7 Offerer Processing of the Answer

The processing of the answer by the offerer is nearly identical to that of the answerer processing the offer. Specifically, the offerer will send STUN requests to the STUN servers listed in the answer. This results in the same connectivity processing, and will also result in the gathering of new STUN derived addresses. The offerer can begin sending media to the answerer immediately (using the address in the m and c lines). Once STUN has verified connectivity of higher priority addresses, media is sent to those addresses instead.

## 5.8 Additional ICE Cycles

After the completion of the offer/answer exchange, both sides may continue to obtain more derived transport addresses. This may occur because a STUN transaction took too long to complete, and thus missed the "window" of the previous offer/answer exchange. Or, it may occur because the previous offer/answer exchange provided additional addresses which resulted in new STUN derived attributes.

At any point when either agent has one or more new gathered

addresses, it MAY initiate a new offer/answer exchange, and a new corresponding ICE cycle. It would add alt attributes to the existing set containing those new gathered addresses. However, if the qvalue of those new gathered addresses is lower than the qvalue for an address that the peer has established connectivity to, the agent SHOULD NOT initiate a new offer/answer exchange just to convey this address. If an offer/answer exchange is taking place anyway (because a higher priority address is available), the lower qvalue addresses SHOULD be included. An agent can determine which addresses a peer has established connectivity to by checking if a STUN request was sent by the peer to that address.

Typically, there won't be more than a small number (2-3) ICE cycles before convergence. Assuming that there is no network packet loss (which can extend the STUN transaction) and zero network latency, it appears that a maximum of two ICE cycles are needed to reach convergence.

## 6. Running STUN on Derived Transport Addresses

One of the seemingly bizarre operations done during the ICE processing is the transmission of a STUN request to a transport address which is obtained through TURN or STUN itself. This actually does work, and in fact, has extremely useful properties. The subsections below go through the detailed operations that would occur at each point to demonstrate correctness and the properties derived from it.

### 6.1 STUN on a TURN Derived Transport Address

Consider a client A that is behind a NAT. It connects to a TURN server on the public side of the NAT. To do that, A binds to a local transport address, say 10.0.1.1:8866, and then sends a TURN request to the TURN server. The NAT translates the net-10 address to 192.0.2.88:5063. Assume that the TURN server is running on 192.0.2.1 and listening for TURN traffic on port 7764. The TURN server allocates a derived transport address 192.0.2.1:26524 to the client, and returns it in the TURN response. Remember that all traffic from the TURN server to the client is sent from 192.0.2.1:7764 to 10.0.1.1:8866.

Now, the client runs a STUN server on 10.0.1.1:8866, and advertises that its server actually runs on 192.0.2.1:26524. Another client, B, sends a STUN request to this server. It sends it from a local transport address, 192.0.2.77:1296. When it arrives at 192.0.2.1:26524, the TURN server "locks down" outgoing traffic, so that data packets received from A are sent to 192.0.2.77:1296. The STUN request is then forwarded to the client, sent with a source address of 192.0.2.1:7764 and a destination address of 192.0.2.88:5063. This passes through the NAT, which rewrites the source address to 10.0.1.1:8866. This arrives at A's STUN server. The server observes the source address of 192.0.2.1:7764, and generates a STUN response containing this value in the MAPPED-ADDRESS attribute. The STUN response is sent with a source address fo 10.0.1.1:8866, and a destination of 192.0.2.1:7764. This arrives at the TURN server, which, because of the lock-down, sends the STUN response with a source address of 192.0.2.1:26524 and destination of 192.0.2.77:1296, which is B's STUN client.

Now, as far as B is concerned, it has obtained a new STUN derived transport address of 192.0.2.1:7764. And indeed, it has! STUN derived transport addresses are scoped to the dialog, so they can only be used by the peer in the dialog. Furthermore, that peer has to send requests from the socket on which the STUN server was running. In this case, A is the peer, and its STUN server was on 10.0.1.1:8866. If it sends to 192.0.2.1:7764, the packet goes to the TURN server,

and due to lock-down, is forwarded to B, and specifically, is forwarded to the transport address B sent the STUN request from. Therefore, the address is indeed a valid STUN derived transport address.

The benefit of this is that it allows two clients to share the same TURN server for media traffic in both directions. With "normal" TURN usage, both clients would obtain a derived address from their own TURN servers. The result is that, for a single call, there are two bindings allocated by each side from their respective servers, and all four are used. With ICE, that drops to two bindings allocated from a single server. Of course, all four bindings are allocated initially. However, once one of the clients begins receiving media on its STUN derived address, it can deallocate its TURN resources.

[[TODO: Include a diagram that shows this pictorially.]]

### 6.2 STUN on a STUN Derived Transport Address

Consider a client A that is behind a NAT. It connects to a STUN server on the public side of the NAT. To do that, A binds to a local transport address, say 10.0.1.1:8866, and then sends a STUN request to the STUN server. The NAT translates the net-10 address to 192.0.2.88:5063. Assume that the STUN server is running on 192.0.2.1 and listening for STUN traffic on port 3478, the default STUN port. The STUN server sees a source IP address of 192.0.2.88:5063, and returns that to the client in the STUN response. The NAT forwards the response to the client.

Now, the client runs a STUN server on 10.0.1.1:8866, and advertises that its server actually runs on 192.0.2.88:5063. Another client, B, sends a STUN request to this address. It sends it from a local transport address, 192.0.2.77:1296. When it arrives at 192.0.2.88:5063 (on the NAT), the NAT rewrites the source address to 10.0.1.1:8866, assuming that it is of the full-cone variety [13], or is restricted, and the permission for 192.0.2.77:1296 is open. This arrives at A's STUN server. The server observes the source address of 192.0.2.77:1296, and generates a STUN response containing this value in the MAPPED-ADDRESS attribute. The STUN response is sent with a source address of 10.0.1.1:8866, and a destination of 192.0.2.77:1296. This arrives at B's STUN client.

Now, as far as B is concerned, it has obtained a new STUN derived transport address of 192.0.2.77:1296. Of course, this is the same address as the local transport address, and therefore this derived address is not used. However, had there been additonal NATs between B and A's NAT, B would end up seeing the binding allocated by that outermost NAT. The net result is that STUN requests sent to a STUN

derived address behave as normal STUN would. However, these STUN requests have the side-effect of creating permissions in the NATs which see those requests in the public to private direction. This turns out to be very useful for traversing restricted NATs.

## 7. SDP Extensions for STUN

A new SDP attribute is defined to support ICE. It is called "alt". The alt attribute MUST be present within a media block of the SDP. It contains an alternative IP address and port (or pair of IP addresses and ports in the case of RTP) that the recipient of the SDP can use instead of the ones indicated in the m and c lines. There MAY be multiple alt attributes in a media block. In that case, each of them MUST contain a different IP address and port (or a differing pair of IP address and ports in the case of RTP).

An agent including an alt attribute in an SDP MUST be prepared to receive STUN requests on the advertised address and port. In the case of RTP, this includes both the RTP and RTCP transport addresses. The username and password that are expected in such STUN requests are advertised using the username and password conveyed in the attribute. It is RECOMMENDED that the agent use different usernames and passwords in each alternate. This allows the agent to know which alternates the peer has been able to establish connectivity to. This knowledge can be used as an optimization to eliminate additional ICE cycles.

The transport addresses in the alt attribute MAY repeat those present in the m and c-lines. In that case, the alt attribute is conveying additional information about the transport addresses in the m and c lines. In particular, it is conveying an identifier for them, an indication of whether the address was peer derived, and a weight indicating a preference for the address. It also implies that the agent is prepared to receive STUN requests on the IP address and port in the m and c lines.

In fact, it is RECOMMENDED that there be an alt attribute for the address and port in the m and c lines.

The syntax of this attribute is:

```
alt-attribute = "alt" ":" id SP qvalue SP derived-from SP
                  username SP password SP
                  unicast-address SP port [unicast-address SP port]
                  ;qvalue from RFC 3261
                  ;unicast-address, port from RFC 2327
usernam       = non-ws-string
passwor       = non-ws-string
id            = token
derived-from  = ":" / id
```

"id" is a unique identifier (unique within the set of alt-attributes

present in offers and answers sent by that agent within the scope of a dialog) that identifies this particular alternative address. "qvalue" is a priority value that indicates the relative preference of this address amongst any others conveyed in alt-attributes in this media stream. The "derived-from" attribute either contains a colon or an id. When it contains a colon, it means that the address is not a peer derived transport address. When it contains an id, it means that the address is a peer derived transport address. The id mirrors the value of i" that identifies the alt-attribute from the peer from which the address was derived. As an example, consider users A and B. User A sends an SDP offer to B, as part of an offer/answer exchange [5]. A indicates, using the alt attribute, that an alternative address exists with "id" 23. B sends a STUN request to this server, and obtains a MAPPED-ADDRESS in the STUN response. This transport address is used in the answer. The m-line containing this address would contain the alt attribute with a "derived-from" of 23.

When a user sends media to a transport address that has "derived-from" containing an id, it MUST do so by sending from the transport address indicated by the id. Continuing the example above, if the transport address provided by A in id 23 was 192.0.2.3:3344, when A sends media to a STUN derived transport address derived from id 23, it MUST send the media from 192.0.2.3:3344. The need for this requirement is described in Section 6.

The "username" and "password" are the username and password that the recipient of the SDP MUST use when connecting to the agent to test connectivity to that alternate. The username and password are scoped to the lifetime of the dialog on which the SDP is being exchanged. If the dialog terminates, the username and password are invalid.

Note that STUN allows both the username and password to contain the space character. However, usernames and passwords used with ICE cannot contain the space.

The security considerations associated with carrying a username and password in the clear in SIP are not as bad as one might think. If an eavesdropper should see the username and password, the worst they can do is send STUN requests to the host. Since STUN is a stateless protocol, the attacker can not alter the processing of the call or otherwise disrupt it. They could flood the server with BindingRequest packets. However, this would be no worse than if the attacker simply floods the host with any kind of packet.

However, integrity protection of the username and password are more important. If an attacker is capable of intercepting the message and modifying the username or password, they could prevent connectivity from being established between peers, and therefore disrupt the call.

Of course, if the attacker can intercept the SIP message, there are many other ways in which they could do that, such as simply discarding the message. Injecting fake SDP with incorect usernames and passwords can also disrupt a call, and does not require the compromise of an intermediate server. A similar attack is possible by modifying most of the SDP attributes. To prevent these kinds of attacks, it is RECOMMENDED that sips be used.

When used with a non-RTP stream, the second address and port MUST NOT be present. In the case of RTP, it MAY be present, in which case it indicates the IP address and port where RTCP is to be sent. When its not present for an RTP stream, it implies that the RTCP packets are sent to the port one higher than the RTP packets.

## 8. Connectivity Preconditions

One of the benefits of ICE is that each side knows with certainty when it is able to communicate with its peer. However, neither side knows for certainty when both sides can communicate with each other. That information represents distributed state spread between both peers. It would be extremely useful to know this piece of information, so that a device can hold off on alerting a user until connectivity has been confirmed. This is exactly the kind of function that SIP preconditions [7] has been designed to provide.

This specification therefore defines the "connected" precondition type. A media stream is considered connected from A to B when connectivity from A to B has been confirmed with STUN for at least one of the alternate transport addresses contained in an "alt" attribute.

```
             A                         B
             |(1) INVITE SDP1          |
             |------------------------>|
             |(2) 183 SDP2             |
             |<------------------------|
             |(3) PRACK                |
             |------------------------>|
             |(4) 200 PRACK            |
             |<------------------------|
             |(5) STUN connectivity    |
             |.........................|
             |(6) UPDATE SDP3          |
             |------------------------>|
             |(7) 200 UPDATE SDP4      |
             |<------------------------|
             |(8) 180 Ringing          |
             |<------------------------|
             |(9) PRACK                |
             |------------------------>|
             |(10) 200 PRACK           |
             |<------------------------|
             |(11) 200 INVITE          |
             |<------------------------|
             |(12) ACK                 |
             |------------------------>|
```

Figure 2

Figure 2 shows a typical call flow used in conjunction with

preconditions. User A does not want the phone to ring unless connectivity is guaranteed. As a result, it sends an INVITE with a connectivity precondition (message 1) that is mandatory in the sendrecv direction. When this arrives at B, B decides to accept the precondition, and therefore generates an answer indicating that the precondition is mandatory in the sendrecv direction. The current status is none, since connectivity hasn't been established in either direction yet. At that point, the ICE cycles begin (which may themselves use UPDATE for the offer/answer exchanges). Assume that B has established connectivity to A. When A establishes connectivity to B, it sends an UPDATE (message 6) with a current status of sendrecv. This meets the precondition. As a result, B's phone rings, causing a 180 to be sent (message 8).

## 9. Example Use Cases

This section contains a number of example use cases. They help to demonstrate that the core ICE algorithm always results in the best connectivity. In all cases, only RTP is shown and discussed, to simplify the discussion. RTCP related operations (generally STUN queries parallel to the RTP ones) are omitted.

The use cases were chosen to represent a superset of those cases described in the current NAT scenarios document [18]. This is to demonstrate that ICE allows for a single solution to be applied to all of the cases described there, and furthermore, that the optimal media flow occurs in all of those cases.

### 9.1 Residential Users

In this scenario, a user has a broadband connection to the Internet, using a cable modem or DSL, for example. In order to provide security, or to run multiple machines, the user has purchased an off-the-shelf "DSL Router" as they are called. These devices, manufactured by companies such as Linksys, Netgear, 2wire, and Netopia, generally include a NAT, simple firewall, DHCP server and client, and a built in ethernet switch of some sort. The firewall generally allows all outgoing traffic, but disallows incoming traffic unless specific port forwarding or a DMZ host has been configured. The NAT treatment of UDP in these boxes varies. The most common types appear to be full-cone and restricted cone.

The user in this scenario wishes to use a communications service from a retail provider, such as net2phone or deltathree, for example. The connection between the user and the provider is through the cable modem or DSL, through the public Internet. The user may have multiple PCs in their home accessing this service, but they are not related in any way. This scenario also includes the case where its not a PC, but a standalone SIP phone. In this case, the provider might be providing some kind of second line VoIP service. This scenario is depicted in Figure 3.

```
                      +---------+    +---------+
                      |Provider|     |TURN/    |
                      | Proxy  |     | STUN    |
                      |        |     | Server  |
                      +----+---+     +----+---+
                           |              |
                           |              |
                         --+--------------+--
                  ///////                    \\\\\\\
```

```
               ///                                 \\\
             ||                                       ||
            |                Internet                   |
           |                                             |
            |                                           |
             ||                                       ||
               \\\                                 ///
                  \\\\\\\                   ///////
                         ---------+---------
                                  | DSL, Cable
                         +--------+-------+
                         |     Home NAT   |
                         +----------------+

                     +--------+       +----------+
                     |        |       |   /  \   |
                     |  PC    |          /SIP \
                     |        |         /Phone \
                     |        |        /        \
                     +--------+       ------------
```

Figure 3: Residence with Single NAT

In this case, the provider administers a SIP proxy and a TURN/STUN server. This server is running STUN on the default port (3478) and TURN on port 5556.

### 9.1.1 Full Cone NAT

```
             A                     As NAT              STUN+TURN Server
             |(1) STUN Bind           |                        |
             |s=10.0.1.1:1010         |                        |
             |d=192.0.2.10:3478       |                        |
             |----------------------->|                        |
             |                        |(2) STUN Bind           |
             |                        |s=192.0.2.1:9988        |
             |                        |d=192.0.2.10:3478       |
             |                        |----------------------->|
             |                        |(3) STUN Resp           |
             |                        |s=192.0.2.10:3478       |
             |                        |d=192.0.2.1:9988        |
             |                        |M=192.0.2.1:9988        |
             |                        |<-----------------------|
             |(4) STUN Resp           |                        |
```

```
             |s=192.0.2.10:3478       |                        |
             |d=10.0.1.1:1010         |                        |
             |M=192.0.2.1:9988        |                        |
             |<-----------------------|                        |
             |(5) STUN Bind           |                        |
             |s=10.0.1.1:1011         |                        |
             |d=192.0.2.10:3478       |                        |
             |----------------------->|                        |
             |                        |(6) STUN Bind           |
             |                        |s=192.0.2.1:9990        |
             |                        |d=192.0.2.10:3478       |
             |                        |----------------------->|
             |                        |(7) STUN Resp           |
             |                        |s=192.0.2.10:3478       |
             |                        |d=192.0.2.1:9990        |
             |                        |M=192.0.2.1:9990        |
             |                        |<-----------------------|
             |(8) STUN Resp           |                        |
             |s=192.0.2.10:3478       |                        |
             |d=10.0.1.1:1011         |                        |
             |M=192.0.2.1:9990        |                        |
             |<-----------------------|                        |
             |(9) TURN Alloc          |                        |
             |s=10.0.1.1:1010         |                        |
             |d=192.0.2.10:5556       |                        |
             |----------------------->|                        |
             |                        |(10) TURN Alloc         |
             |                        |s=192.0.2.1:9988        |
             |                        |d=192.0.2.10:5556       |
             |                        |----------------------->|
             |                        |(11) TURN Resp          |
             |                        |s=192.0.2.10:5556       |
             |                        |d=192.0.1.1:9988        |
             |                        |M=192.0.2.10:8076       |
             |                        |<-----------------------|
             |(12) TURN Resp          |                        |
             |s=192.0.2.10:5556       |                        |
             |d=10.0.1.1:1010         |                        |
             |M=192.0.2.10:8076       |                        |
             |<-----------------------|                        |
             |(13) TURN Alloc         |                        |
             |s=10.0.1.1:1011         |                        |
             |d=192.0.2.10:5556       |                        |
             |----------------------->|                        |
             |                        |(14) TURN Alloc         |
             |                        |s=192.0.2.1:9990        |
             |                        |d=192.0.2.10:5556       |
             |                        |----------------------->|
```

```
             |                        |(15) TURN Resp         |
             |                        |s=192.0.2.10:5556      |
             |                        |d=192.0.1.1:9990       |
             |                        |M=192.0.2.10:8077      |
             |                        |<----------------------|
             |(16) TURN Resp          |                       |
             |s=192.0.2.10:5556       |                       |
             |d=10.0.1.1:1011         |                       |
             |M=192.0.2.10:8077       |                       |
             |<-----------------------|                       |
```

Figure 4: Message sequence for A's Unilateral Allocations

We first consider the case where two such residential users call each other, and both are using NATs of the full-cone variety. The caller follows the ICE algorithm. As such, it firsts allocates a pair of ports on its local interface for RTP and RTCP traffic (10.0.1.1:1010 and 10.0.1.1:1011). As shown in Figure 4, the client issues a STUN request from the RTP port (message 1), which passes through the NAT on its way to the STUN server. In the figure, the "s=" indicates the source transport address of the message, and "d=" indicates the destination transport address. The NAT translates the 10.0.1.1:1010 to 192.0.2.1:9988, and this request arrives at the STUN server (message 2). The STUN server copies the source address into the MAPPED-ADDRESS field in the STUN response (the M= line in message 3), and this passes through the NAT, back to the client. The client now has a STUN derived transport address of 192.2.0.1:9988. It follows a similar process for RTCP (messages 5-8). This STUN derived transport address, 192.0.2.1:9990 is not adjacent to the RTP port.

Next, the client follows a similar process to obtain a TURN port for RTP (messages 9-12) and RTCP (messages 13-16). The TURN requests are also sent from the same local transport address. Note, however, that the TURN derived transport addresses for RTP (192.0.2.10:8076) and RTCP (192.0.2.10:8077) are on adjacent ports. This is because the TURN pre-allocation procedure was used in the TURN request for the RTP port (message 9).

The client prioritizes these addresses, choosing the local interface address with priority 1.0, the STUN address with priority 0.8, and the TURN address with priority 0.4. From this, it generates an offer that looks like this:

```
v=0
o=alice 2890844730 2890844731 IN IP4 host.example.com
s=
c=IN IP4 192.0.2.10
t=0 0
m=audio 8076 RTP/AVP 0
a=alt:1 1.0 : user 9kksj== 10.0.1.1 1010
a=alt:2 0.8 : user1 9kksk== 192.0.2.1 9988 192.0.2.1 9990
a=alt:3 0.4 : user2 9kksl== 192.0.2.10 8076
```

Figure 5: A's Offer

Note how the TURN derived transport address is used in the m and c lines, since this is the address with the highest probability of working with a non-ICE peer. That address is also included in the list of alteratives (with ID 3). Also note that because the STUN derived transport address for RTP and RTCP were not adjacent, two transport addresses are provided for alternate 2.

```
B                     Bs NAT              STUN+TURN Server
|(1) STUN Bind             |                         |
|s=192.168.3.1:23766       |                         |
|d=192.0.2.10:3478         |                         |
|------------------------->|                         |
|                          |(2) STUN Bind            |
|                          |s=192.0.2.2:10892        |
|                          |d=192.0.2.10:3478        |
|                          |------------------------>|
|                          |(3) STUN Resp            |
|                          |s=192.0.2.10:3478        |
|                          |d=192.0.2.2:10892        |
|                          |M=192.0.2.2:10892        |
|                          |<------------------------|
|(4) STUN Resp             |                         |
|s=192.0.2.10:3478         |                         |
|d=192.168.3.1:23766       |                         |
|M=192.0.2.2:10892         |                         |
|<-------------------------|                         |
|(5) STUN Bind             |                         |
|s=192.168.3.1:23767       |                         |
|d=192.0.2.10:3478         |                         |
|------------------------->|                         |
|                          |(6) STUN Bind            |
|                          |s=192.0.2.2:10893        |
|                          |d=192.0.2.10:3478        |
|                          |------------------------>|
|                          |(7) STUN Resp            |
```

```
             |                        |s=192.0.2.10:3478       |
             |                        |d=192.0.2.2:10893       |
             |                        |M=192.0.2.2:10893       |
             |                        |<-----------------------|
             |(8) STUN Resp           |                        |
             |s=192.0.2.10:3478       |                        |
             |d=192.168.3.1:23767     |                        |
             |M=192.0.2.2:10893       |                        |
             |<-----------------------|                        |
             |(9) TURN Alloc          |                        |
             |s=192.168.3.1:23766     |                        |
             |d=192.0.2.10:5556       |                        |
             |----------------------->|                        |
             |                        |(10) TURN Alloc         |
             |                        |s=192.0.2.2:10892       |
             |                        |d=192.0.2.10:5556       |
             |                        |----------------------->|
             |                        |(11) TURN Resp          |
             |                        |s=192.0.2.10:5556       |
             |                        |d=192.0.2.2:10892       |
             |                        |M=192.0.2.10:8078       |
             |                        |<-----------------------|
             |(12) TURN Resp          |                        |
             |s=192.0.2.10:5556       |                        |
             |d=192.168.3.1:23766     |                        |
             |M=192.0.2.10:8078       |                        |
             |<-----------------------|                        |
             |(13) TURN Alloc         |                        |
             |s=192.168.3.1:23767     |                        |
             |d=192.0.2.10:5556       |                        |
             |----------------------->|                        |
             |                        |(14) TURN Alloc         |
             |                        |s=192.0.2.2:10893       |
             |                        |d=192.0.2.10:5556       |
             |                        |----------------------->|
             |                        |(15) TURN Resp          |
             |                        |s=192.0.2.10:5556       |
             |                        |d=192.0.2.2:10893       |
             |                        |M=192.0.2.10:8079       |
             |                        |<-----------------------|
             |(16) TURN Resp          |                        |
             |s=192.0.2.10:5556       |                        |
             |d=192.168.3.1:23767     |                        |
             |M=192.0.2.10:8079       |                        |
             |<-----------------------|                        |
```

Figure 6: Message sequence for B's Unilateral Allocations

This offer arrives at the called party, user B. User B is also behind a full-cone NAT, and is using the 192.168/16 private address space internally. It happens to be using the same service provider as A, and is therefore using the same TURN server, at 192.0.2.10:5556. User B follows the same set of procedures followed by user A. It uses local interfaces, STUN, and TURN, and obtains a set of transport addresses that it can use. This process is shown in Figure 6. This process differs from that of Figure 4 only in the actual addresses and ports used and obtained.

```
             A          As NAT  TURN + STUN Server  Bs NAT           B
             |             |             |             |(1) STUN Bind|
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=10.0.1.1:1010
             |             |             |             |<------------|
             |             |             |Unreachable  |             |
             |             |             |             |(2) STUN Bind|
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.1:9988
             |             |             |             |<------------|
             |             |(3) STUN Bind|             |             |
             |             |s=192.0.2.2:10892          |             |
             |             |d=192.0.2.1:9988           |             |
             |             |<--------------------------|             |
             |(4) STUN Bind|             |             |             |
             |s=192.0.2.2:10892          |             |             |
             |d=10.0.1.1:1010            |             |             |
             |<------------|             |             |             |
             |(5) STUN Reply             |             |             |
             |s=10.0.1.1:1010            |             |             |
             |d=192.0.2.2:10892          |             |             |
             |M=192.0.2.2:10892          |             |             |
             |------------>|             |             |             |
             |             |(6) STUN Reply             |             |
             |             |s=192.0.2.1:9988           |             |
             |             |d=192.0.2.2:10892          |             |
             |             |M=192.0.2.2:10892          |             |
             |             |-------------------------->|             |
             |             |             |             |(7) STUN Reply
             |             |             |             |s=192.0.2.1:9988
             |             |             |             |d=192.168.3.1:23766
             |             |             |             |M=192.0.2.2:10892
             |             |             |             |------------>|
             |             |             |             |(8) STUN Bind|
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.10:8076
             |             |             |             |<------------|
```

```
          |             |             |(9) STUN Bind|             |
          |             |             |s=192.0.2.2:10892          |
          |             |             |d=192.0.2.10:8076          |
          |             |             |<------------|             |
          |             |(10) STUN Bind             |             |
          |             |s=192.0.2.10:5556          |             |
          |             |d=192.0.2.1:9988           |             |
          |             |<------------|             |             |
          |(11) STUN Bind             |             |             |
          |s=192.0.2.10:5556          |             |             |
          |d=10.0.1.1:1010            |             |             |
          |<------------|             |             |             |
          |(12) STUN Reply            |             |             |
          |s=10.0.1.1:1010            |             |             |
          |d=192.0.2.10:5556          |             |             |
          |M=192.0.2.10:5556          |             |             |
          |------------>|             |             |             |
          |             |(13) STUN Reply            |             |
          |             |s=192.0.2.1:9988           |             |
          |             |d=192.0.2.10:5556          |             |
          |             |M=192.0.2.10:5556          |             |
          |             |------------>|             |             |
          |             |             |(14) STUN Reply            |
          |             |             |s=192.0.2.10:8076          |
          |             |             |d=192.0.2.2:10892          |
          |             |             |M=192.0.2.10:5556          |
          |             |             |------------>|             |
          |             |             |             |(15) STUN Reply
          |             |             |             |s=192.0.2.10:8076
          |             |             |             |d=192.168.3.1:23766
          |             |             |             |M=192.0.2.10:5556
          |             |             |             |------------>|
```

Figure 7: B's Connectivity Checks

While B's phone is ringing, B's user agent uses STUN to test connectivity from its local transport address pair (192.168.3.1:23766 and 192.168.3.1:23767) to the three alternates listed in the offer. The flow for that is shown in Figure 7. This flow, and the discussions, only consider the RTP transport addresses. The procedures would all be identical for RTCP. First, B tests connectivity to the alternate with ID 1, which is 10.0.1.1:1010. It does so by attempting to send a STUN request to this address (message 1). Of course, this is a private address, and not in the same network as B. Therefore, it is unreachable, and no STUN response is received.

In parallel, B tests connectivity to the alternate with ID 2, which is 192.0.2.1:9988. To do this, it sends a STUN request to that

address. It sends it with a source address equal to its local transport address; the same one that it used to send the previous TURN and STUN packets (192.168.3.1:23766). This request (message 2) arrives at the NAT. Since the NAT is full cone, and since this address has an existing binding, the NAT translates the source address to that existing binding, 192.0.2.2:10892. This request (message 3) continues onwards to A's NAT. Since A's NAT is also full cone, the existing binding for 192.0.2.1:9988 is used, and the destination address is translated to 10.0.1.1:1010 and then forwarded towards A (message 4). A receives this. It verifies the username and password, and then generates a response. The response contains a MAPPED-ADDRESS equal to the source address seen in the STUN request (192.0.2.2:10892). It passes back through A's NAT (message 5), through B's NAT (message 6), and back to B (message 7).

B examines the MAPPED-ADDRESS in the STUN response. Its 192.0.2.2:10892. However, this is not a new address. B is already aware of this address as a result of its initial STUN Binding requests to the TURN/STUN server (Figure 6). As such, no additional addresses were learned.

In parallel with the tests against ID 2, B tests connectivity to the alternate with ID 3. This is the address A allocated through TURN. Of course, B does not know this. B sends a STUN request to this address (192.0.2.10:8076), and sends it from the same local transport address (192.168.3.1:23766) (message 8). The NAT, once again, translates the source address to 192.0.2.2:10892 (message 9). This is routed to the TURN server. The TURN server locks down the binding allocated to A, such that it will now begin relaying packets sent from A to 192.0.2.2:10892. The TURN server forwards the packet towards A (message 10). This reaches A's NAT, which translates the destination address based on the existing binding. The STUN request is then delivered to A (message 11). A verifies the username and password, and then generates a STUN response. This response contains the source address that the request came from. In this case, that source address is the public transport address of the TURN server (192.0.2.10:5556). This STUN response is relayed all the way back to B (messages 12-15).

B examines the MAPPED-ADDRESS in this STUN response. It's 192.0.2.10:5556, which is a new address. As a result, B has now obtained a peer derived STUN address. It adds this to its list of transport addresses. Its priority equals that of the address it was derived from - ID 3 - which has a qvalue of 0.4.

At some point, B picks up, and an answer is generated. The answer would look like this:

```
v=0
o=bob 2890844730 289084871 IN IP4 host2.example.com
s=
c=IN IP4 192.0.2.10
t=0 0
m=audio 8078 RTP/AVP 0
a=alt:4 1.0 : peer as88jl 192.168.3.1 23766
a=alt:5 0.8 : peer1 as88kl 192.0.2.2 10892
a=alt:6 0.4 : peer2 as88ll 192.0.2.10 8078
a=alt:7 0.4 3 peer3 as88ml 192.0.2.10 5556
```

Figure 8: B's Answer

Note how the alternative with ID 7 indicates that it was derived from the alternate with ID 3. Also, note that the four alternates use different IDs than the ones from the offer. This is for readability purposes only. The IDs are scoped to that specific agent, and there is no requirement that they do not use the same values.

This answer is sent to A. At the same time, B can send audio to A using the highest priority alternate that connectivity was established to. That is the alternate with ID 2, A's STUN derived transport address.

```
A          As NAT  TURN + STUN Server  Bs NAT           B
|(1) STUN Bind|             |             |             |
|s=10.0.1.1:1010            |             |             |
|d=192.168.3.1:23766        |             |             |
|------------>|             |             |             |
|             |Unreachable  |             |             |
|(2) STUN Bind|             |             |             |
|s=10.0.1.1:1010            |             |             |
|d=192.0.2.2:10892          |             |             |
|------------>|             |             |             |
|             |(3) STUN Bind|             |             |
|             |s=192.0.2.1:9988           |             |
|             |d=192.0.2.2:10892          |             |
|             |-------------------------->|             |
|             |             |             |(4) STUN Bind|
|             |             |             |s=192.0.2.1:9988
|             |             |             |d=192.168.3.1:23766
|             |             |             |------------>|
|             |             |             |(5) STUN Reply
|             |             |             |s=192.168.3.1:23766
|             |             |             |d=192.0.2.1:9988
|             |             |             |M=192.0.2.1:9988
```

```
             |             |             |             |<------------|
             |             |(6) STUN Reply             |             |
             |             |s=192.0.2.2:10892          |             |
             |             |d=192.0.2.1:9988           |             |
             |             |M=192.0.2.1:9988           |             |
             |             |<--------------------------|             |
             |(7) STUN Reply             |             |             |
             |s=192.0.2.2:10892          |             |             |
             |d=10.0.1.1:1010            |             |             |
             |M=192.0.2.1:9988           |             |             |
             |<------------|             |             |             |
             |(8) STUN Bind|             |             |             |
             |s=10.0.1.1:1010            |             |             |
             |d=192.0.2.10:8078          |             |             |
             |------------>|             |             |             |
             |             |(9) STUN Bind|             |             |
             |             |s=192.0.2.1:9988           |             |
             |             |d=192.0.2.10:8078          |             |
             |             |------------>|             |             |
             |             |             |(10) STUN Bind             |
             |             |             |s=192.0.2.10:5556          |
             |             |             |d=192.0.2.2:10892          |
             |             |             |------------>|             |
             |             |             |             |(11) STUN Bind
             |             |             |             |s=192.0.2.10:5556
             |             |             |             |d=192.168.3.1:23766
             |             |             |             |------------>|
             |             |             |             |(12) STUN Reply
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.10:5556
             |             |             |             |M=192.0.2.10:5556
             |             |             |             |<------------|
             |             |             |(13) STUN Reply            |
             |             |             |s=192.0.2.2:10892          |
             |             |             |d=192.0.2.10:5556          |
             |             |             |M=192.0.2.10:5556          |
             |             |             |<------------|             |
             |             |(14) STUN Reply            |             |
             |             |s=192.0.2.10:8078          |             |
             |             |d=192.0.2.1:9988           |             |
             |             |M=192.0.2.10:5556          |             |
             |             |<------------|             |             |
             |(15) STUN Reply            |             |             |
             |s=192.0.2.10:8078          |             |             |
             |d=10.0.1.1:1010            |             |             |
             |M=192.0.2.10:5556          |             |             |
             |<------------|             |             |             |
             |(16) STUN Bind             |             |             |
```

```
             |s=10.0.1.1:1010            |             |             |
             |d=192.0.2.10:5556          |             |             |
             |------------>|             |             |             |
             |             |(17) STUN Bind             |             |
             |             |s=192.0.2.1:9988           |             |
             |             |d=192.0.2.10:5556          |             |
             |             |------------>|             |             |
             |             |             |(18) STUN Bind             |
             |             |             |s=192.0.2.10:8076          |
             |             |             |d=192.0.2.2:10892          |
             |             |             |------------>|             |
             |             |             |             |(19) STUN Bind
             |             |             |             |s=192.0.2.10:8076
             |             |             |             |d=192.168.3.1:23766
             |             |             |             |------------>|
             |             |             |             |(20) STUN Reply
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.10:8076
             |             |             |             |M=192.0.2.10:8076
             |             |             |             |<------------|
             |             |             |(21) STUN Reply            |
             |             |             |s=192.0.2.2:10892          |
             |             |             |d=192.0.2.10:8076          |
             |             |             |M=192.0.2.10:8076          |
             |             |             |<------------|             |
             |             |(22) STUN Reply            |             |
             |             |s=192.0.2.10:5556          |             |
             |             |d=192.0.2.1:9988           |             |
             |             |M=192.0.2.10:8076          |             |
             |             |<------------|             |             |
             |(23) STUN Reply            |             |             |
             |s=192.0.2.10:5556          |             |             |
             |d=10.0.1.1:1010            |             |             |
             |M=192.0.2.10:8076          |             |             |
             |<------------|             |             |             |
```

Figure 9: A's Connectivity Checks

When the answer arrives at A, A performs similar connectivity checks, shown in Figure 9. Each connectivity check is a STUN request sent from its local transport address (10.0.1.1:1010). The first is to the alternate with ID 4, which is 192.168.3.1:23766. The STUN request to this address (message 1) fails, since this is an unreachable private address. The second check is to the alternate with ID 5 (192.0.2.2:10892), which is the public address for B obtained as a result of STUN requests to the network server. Messages 2-7 represent the flow for this case. It is similar to the sequence in Figure 7 messages 2-7, differing only in the IP addresses. The result of this

check provides a peer derived transport address of 192.0.2.1:9988. A already knows this address. The third connectivity check is to the alternate with ID 6 (192.0.2.10:8078). This represents A's TURN derived transport address. Messages 8-15 represent the check for this address, and they are also similar to messages 8-15 of Figure 7. This check provides A with a peer derived transport address of 192.0.2.10:5556. This represents a new address for A. It has a priority equal to the address it was derived from, which is 0.4.

The final connectivity check is to the alternate with ID 7 (192.0.2.10 5556). The SDP indicates that this address itself is a peer derived transport address. It was derived from A's transport address with ID 3, which is 192.0.2.10:8076, its TURN derived transport address. Because of that, the STUN request is sent from the local transport address that 192.0.2.10:8076 was derived from. This local address is 10.0.1.1:1010. The message sequence for this check is represented by messages 16-23 of Figure 9. The STUN request is sent with a source address of 10.0.1.1:1010, to 192.0.2.10:5556. This is the well-known address of the TURN relay. This message passes through the NAT, and the source address is translated to A's public address, 192.0.2.1:9988 (message 17). Note that this same public address is used for all requests sent from 10.0.1.1:1010 because the NAT is full-cone. This arrives at the TURN server. The TURN server associates this message (which is just an arbitrary UDP packet as far as the TURN server is concerned) with the binding created for A. The peer in this case has been locked down. So, the packet is forwarded with a source address equal to the binding allocated to A (192.0.2.10:8076) and a destination address equal to the locked-down address (192.0.2.2:10892) (message 18). This arrives at B's NAT, where the destination address is translated to B's private address, 192.168.3.1:23766 (message 19). This arrives at B, which notes the source address in the STUN reply (192.0.2.10:8076). This reply is forwarded back to A (messages 20-23). From this, A sees a peer derived transport address of 192.0.2.10:8076. However, it already knows this address.

The result of the connectivity checks is that A determines it has connectivity to the alternates with IDs 5, 6 and 7. Of these, the one with ID 5 has the highest priority, and so this one is used to send media. Of course, A could have been sending media to B during these tests using the address in the m and c lines, which represents B's TURN derived transport address. Once the connectivity checks complete, A can switch to the one with ID 5, which is B's STUN derived transport address.

The connectivity checks also provided A with a new peer derived transport address - 192.0.2.10:5556 - with a priority of 0.4. However, A had received STUN requests on its alternates with IDs 2

and 3. The one with ID 2 (its STUN derived transport address) has higher priority than 0.4. So, A knows that generating a new ICE cycle to convey this address would not be useful. Thus, no new offer is sent. Indeed, since A had received a STUN request from B on its STUN derived transport address, A knows that its lower priority derived transport address is no longer needed. So, it is able to free up the TURN derived transport address a few seconds later. The same goes for B. Once it receives the STUN request to its TURN derived transport address (message 11 of Figure 9, it can free its TURN derived transport address.

In conclusion, the result in this case is that A and B will communicate with each other using their STUN derived transport addresses.

### 9.1.2 Symmetric NAT

```
             A                      As NAT              STUN+TURN Server
             |(1) STUN Bind           |                         |
             |s=10.0.1.1:1010         |                         |
             |d=192.0.2.10:3478       |                         |
             |----------------------->|                         |
             |                        |(2) STUN Bind            |
             |                        |s=192.0.2.1:9988         |
             |                        |d=192.0.2.10:3478        |
             |                        |------------------------>|
             |                        |(3) STUN Resp            |
             |                        |s=192.0.2.10:3478        |
             |                        |d=192.0.2.1:9988         |
             |                        |M=192.0.2.1:9988         |
             |                        |<------------------------|
             |(4) STUN Resp           |                         |
             |s=192.0.2.10:3478       |                         |
             |d=10.0.1.1:1010         |                         |
             |M=192.0.2.1:9988        |                         |
             |<-----------------------|                         |
             |(5) STUN Bind           |                         |
             |s=10.0.1.1:1011         |                         |
             |d=192.0.2.10:3478       |                         |
             |----------------------->|                         |
             |                        |(6) STUN Bind            |
             |                        |s=192.0.2.1:9990         |
             |                        |d=192.0.2.10:3478        |
             |                        |------------------------>|
             |                        |(7) STUN Resp            |
             |                        |s=192.0.2.10:3478        |
             |                        |d=192.0.2.1:9990         |
```

```
             |                        |M=192.0.2.1:9990        |
             |                        |<-----------------------|
             |(8) STUN Resp           |                        |
             |s=192.0.2.10:3478       |                        |
             |d=10.0.1.1:1011         |                        |
             |M=192.0.2.1:9990        |                        |
             |<-----------------------|                        |
             |(9) TURN Alloc          |                        |
             |s=10.0.1.1:1010         |                        |
             |d=192.0.2.10:5556       |                        |
             |----------------------->|                        |
             |                        |(10) TURN Alloc         |
             |                        |s=192.0.2.1:9991        |
             |                        |d=192.0.2.10:5556       |
             |                        |----------------------->|
             |                        |(11) TURN Resp          |
             |                        |s=192.0.2.10:5556       |
             |                        |d=192.0.1.1:9991        |
             |                        |M=192.0.2.10:8076       |
             |                        |<-----------------------|
             |(12) TURN Resp          |                        |
             |s=192.0.2.10:5556       |                        |
             |d=10.0.1.1:1010         |                        |
             |M=192.0.2.10:8076       |                        |
             |<-----------------------|                        |
             |(13) TURN Alloc         |                        |
             |s=10.0.1.1:1011         |                        |
             |d=192.0.2.10:5556       |                        |
             |----------------------->|                        |
             |                        |(14) TURN Alloc         |
             |                        |s=192.0.2.1:9992        |
             |                        |d=192.0.2.10:5556       |
             |                        |----------------------->|
             |                        |(15) TURN Resp          |
             |                        |s=192.0.2.10:5556       |
             |                        |d=192.0.1.1:9992        |
             |                        |M=192.0.2.10:8077       |
             |                        |<-----------------------|
             |(16) TURN Resp          |                        |
             |s=192.0.2.10:5556       |                        |
             |d=10.0.1.1:1011         |                        |
             |M=192.0.2.10:8077       |                        |
             |<-----------------------|                        |
```

Figure 10: A's Unilateral Allocations

In this case, both residential users have symmetric NATs. The call starts again with A performing its unilateral allocations, as is

shown in Figure 10. This message sequence is nearly identical to that of Figure 4. The only difference is that, because the NAT is symmetric, different bindings are allocated for the two STUN and two TURN queries. A's discovers an identical set of addresses, however, and so generates the same offer as in Figure 5.

```
             B                     Bs NAT              STUN+TURN Server
             |(1) STUN Bind           |                         |
             |s=192.168.3.1:23766     |                         |
             |d=192.0.2.10:3478       |                         |
             |----------------------->|                         |
             |                        |(2) STUN Bind            |
             |                        |s=192.0.2.2:10892        |
             |                        |d=192.0.2.10:3478        |
             |                        |------------------------>|
             |                        |(3) STUN Resp            |
             |                        |s=192.0.2.10:3478        |
             |                        |d=192.0.2.2:10892        |
             |                        |M=192.0.2.2:10892        |
             |                        |<------------------------|
             |(4) STUN Resp           |                         |
             |s=192.0.2.10:3478       |                         |
             |d=192.168.3.1:23766     |                         |
             |M=192.0.2.2:10892       |                         |
             |<-----------------------|                         |
             |(5) STUN Bind           |                         |
             |s=192.168.3.1:23767     |                         |
             |d=192.0.2.10:3478       |                         |
             |----------------------->|                         |
             |                        |(6) STUN Bind            |
             |                        |s=192.0.2.2:10893        |
             |                        |d=192.0.2.10:3478        |
             |                        |------------------------>|
             |                        |(7) STUN Resp            |
             |                        |s=192.0.2.10:3478        |
             |                        |d=192.0.2.2:10893        |
             |                        |M=192.0.2.2:10893        |
             |                        |<------------------------|
             |(8) STUN Resp           |                         |
             |s=192.0.2.10:3478       |                         |
             |d=192.168.3.1:23767     |                         |
             |M=192.0.2.2:10893       |                         |
             |<-----------------------|                         |
             |(9) TURN Alloc          |                         |
             |s=192.168.3.1:23766     |                         |
             |d=192.0.2.10:5556       |                         |
             |----------------------->|                         |
```

```
              |                        |(10) TURN Alloc         |
              |                        |s=192.0.2.2:10894       |
              |                        |d=192.0.2.10:5556       |
              |                        |----------------------->|
              |                        |(11) TURN Resp          |
              |                        |s=192.0.2.10:5556       |
              |                        |d=192.0.2.2:10894       |
              |                        |M=192.0.2.10:8078       |
              |                        |<-----------------------|
              |(12) TURN Resp          |                        |
              |s=192.0.2.10:5556       |                        |
              |d=192.168.3.1:23766     |                        |
              |M=192.0.2.10:8078       |                        |
              |<-----------------------|                        |
              |(13) TURN Alloc         |                        |
              |s=192.168.3.1:23767     |                        |
              |d=192.0.2.10:5556       |                        |
              |----------------------->|                        |
              |                        |(14) TURN Alloc         |
              |                        |s=192.0.2.2:10895       |
              |                        |d=192.0.2.10:5556       |
              |                        |----------------------->|
              |                        |(15) TURN Resp          |
              |                        |s=192.0.2.10:5556       |
              |                        |d=192.0.2.2:10895       |
              |                        |M=192.0.2.10:8079       |
              |                        |<-----------------------|
              |(16) TURN Resp          |                        |
              |s=192.0.2.10:5556       |                        |
              |d=192.168.3.1:23767     |                        |
              |M=192.0.2.10:8079       |                        |
              |<-----------------------|                        |
```

Figure 11: B's Unilateral Allocations

When B receives this offer, it performs its unilateral allocations. Like A's, these allocations (shown in Figure 11) are almost identical to those in Figure 6. They differ in the same way - the NAT will allocate a different binding for each of the two STUN and two TURN queries. However, the set of derived transport address is the same. B now begins performing connectivity checks. These are shown in Figure 12. As in the previous case (Figure 7), the STUN request to 10.0.1.1:1010 fails. However, here, the STUN request to 192.0.2.1:9988 also fails. Thats because this packet arrives at A's NAT, and the NAT finds that the public transport address 192.0.2.1:9988 has been allocated, however, it was allocated when the client sent to 192.0.2.10:3478. Here, the source address is not 192.0.2.10:3478, and so the packet is discarded. The STUN request to

192.0.2.10:8076 does work, however. Thats because the TURN server sends the request from the same IP address and port that it received the original TURN allocation request on.

```
             A          As NAT  TURN + STUN Server  Bs NAT           B
             |             |             |             |(1) STUN Bind|
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=10.0.1.1:1010
             |             |             |             |<------------|
             |             |             |Unreachable  |             |
             |             |             |             |(2) STUN Bind|
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.1:9988
             |             |             |             |<------------|
             |             |(3) STUN Bind|             |             |
             |             |s=192.0.2.2:10896           |             |
             |             |d=192.0.2.1:9988            |             |
             |             |<--------------------------|             |
             |             |Unreachable  |             |             |
             |             |             |             |(4) STUN Bind|
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.10:8076
             |             |             |             |<------------|
             |             |             |(5) STUN Bind|             |
             |             |             |s=192.0.2.2:10897           |
             |             |             |d=192.0.2.10:8076          |
             |             |             |<------------|             |
             |             |(6) STUN Bind|             |             |
             |             |s=192.0.2.10:5556          |             |
             |             |d=192.0.2.1:9991           |             |
             |             |<------------|             |             |
             |(7) STUN Bind|             |             |             |
             |s=192.0.2.10:5556          |             |             |
             |d=10.0.1.1:1010            |             |             |
             |<------------|             |             |             |
             |(8) STUN Reply             |             |             |
             |s=10.0.1.1:1010            |             |             |
             |d=192.0.2.10:5556          |             |             |
             |M=192.0.2.10:5556          |             |             |
             |------------>|             |             |             |
             |             |(9) STUN Reply             |             |
             |             |s=192.0.2.1:9991           |             |
             |             |d=192.0.2.10:5556          |             |
             |             |M=192.0.2.10:5556          |             |
             |             |------------>|             |             |
             |             |             |(10) STUN Reply            |
             |             |             |s=192.0.2.10:8076          |
```

```
             |             |             |d=192.0.2.2:10897          |
             |             |             |M=192.0.2.10:5556          |
             |             |             |------------>|             |
             |             |             |             |(11) STUN Reply
             |             |             |             |s=192.0.2.10:8076
             |             |             |             |d=192.168.3.1:23766
             |             |             |             |M=192.0.2.10:5556
             |             |             |             |------------>|
```

Figure 12: B's Connectivity Checks

B's answer to A is the same as in Figure 8. However, B has only established connectivity to A's TURN derived transport address, and so it sends media there.

```
             A          As NAT  TURN + STUN Server  Bs NAT           B
             |(1) STUN Bind|             |             |             |
             |s=10.0.1.1:1010            |             |             |
             |d=192.168.3.1:23766        |             |             |
             |------------>|             |             |             |
             |             |Unreachable  |             |             |
             |(2) STUN Bind|             |             |             |
             |s=10.0.1.1:1010            |             |             |
             |d=192.0.2.2:10892          |             |             |
             |------------>|             |             |             |
             |             |(3) STUN Bind|             |             |
             |             |s=192.0.2.1:9993           |             |
             |             |d=192.0.2.2:10892          |             |
             |             |-------------------------->|             |
             |             |             |             |Unreachable  |
             |(4) STUN Bind|             |             |             |
             |s=10.0.1.1:1010            |             |             |
             |d=192.0.2.10:8078          |             |             |
             |------------>|             |             |             |
             |             |(5) STUN Bind|             |             |
             |             |s=192.0.2.1:9994           |             |
             |             |d=192.0.2.10:8078          |             |
             |             |------------>|             |             |
             |             |             |(6) STUN Bind|             |
             |             |             |s=192.0.2.10:5556          |
             |             |             |d=192.0.2.2:10894          |
             |             |             |------------>|             |
             |             |             |             |(7) STUN Bind|
             |             |             |             |s=192.0.2.10:5556
             |             |             |             |d=192.168.3.1:23766
             |             |             |             |------------>|
             |             |             |             |(8) STUN Reply
```

```
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.10:5556
             |             |             |             |M=192.0.2.10:5556
             |             |             |             |<------------|
             |             |             |(9) STUN Reply             |
             |             |             |s=192.0.2.2:10894          |
             |             |             |d=192.0.2.10:5556          |
             |             |             |M=192.0.2.10:5556          |
             |             |             |<------------|             |
             |             |(10) STUN Reply            |             |
             |             |s=192.0.2.10:8078          |             |
             |             |d=192.0.2.1:9994           |             |
             |             |M=192.0.2.10:5556          |             |
             |             |<------------|             |             |
             |(11) STUN Reply            |             |             |
             |s=192.0.2.10:8078          |             |             |
             |d=10.0.1.1:1010            |             |             |
             |M=192.0.2.10:5556          |             |             |
             |<------------|             |             |             |
             |(12) STUN Bind             |             |             |
             |s=10.0.1.1:1010            |             |             |
             |d=192.0.2.10:5556          |             |             |
             |------------>|             |             |             |
             |             |(13) STUN Bind             |             |
             |             |s=192.0.2.1:9991           |             |
             |             |d=192.0.2.10:5556          |             |
             |             |------------>|             |             |
             |             |             |(14) STUN Bind             |
             |             |             |s=192.0.2.10:8076          |
             |             |             |d=192.0.2.2:10897          |
             |             |             |------------>|             |
             |             |             |             |(15) STUN Bind
             |             |             |             |s=192.0.2.10:8076
             |             |             |             |d=192.168.3.1:23766
             |             |             |             |------------>|
             |             |             |             |(16) STUN Reply
             |             |             |             |s=192.168.3.1:23766
             |             |             |             |d=192.0.2.10:8076
             |             |             |             |M=192.0.2.10:8076
             |             |             |             |<------------|
             |             |             |(17) STUN Reply            |
             |             |             |s=192.0.2.2:10897          |
             |             |             |d=192.0.2.10:8076          |
             |             |             |M=192.0.2.10:8076          |
             |             |             |<------------|             |
             |             |(18) STUN Reply            |             |
             |             |s=192.0.2.10:5556          |             |
             |             |d=192.0.2.1:9991           |             |
```

```
             |             |M=192.0.2.10:8076           |             |
             |             |<------------|              |             |
             |(19) STUN Reply            |              |             |
             |s=192.0.2.10:5556          |              |             |
             |d=10.0.1.1:1010            |              |             |
             |M=192.0.2.10:8076          |              |             |
             |<------------|             |              |             |
```

Figure 13: A's Connectivity Checks

When A gets the answer, it too performs its connectivity checks, as shown in Figure 13. As expected, the connectivity checks to B's private address and its STUN derived transport addresses fail. The checks to B's TURN derived transport address succeeds, as does the check to B's peer derived transport address. Both have a qvalue of 0.4. However, a peer-derived address is always preferred. So, A will send media to B using 192.0.2.10:5556, which will reach B as a result of the lock-down on its own TURN binding. As in the full-cone case, A won't bother to perform another offer with the new peer derived transport address it learned from message 19 (192.0.2.10:5556), since it knows that this is not of higher priority than ones that B has already connected to.

Once A connects to B's peer derived address (messages 12 to 19 in Figure 13), B knows that its equal priority TURN derived transport address won't be used, so it can free it.

OPEN ISSUE: The same argument can be made about A, in which case both sides would free their TURN addresses, and nothing works. Need to come up with a sane prioritization so it doesnt happen.

## 9.2 Enterprise

```
                                Public Internet


                                192.0.2.1
                             +---------+
                             |         |
       ----------------------| Firewall|--------------------------
                             |  NAT    |
        10.0.0.0/16          +---------+                 DMZ



                 +---------+              +---------+
```

```
                 |         |              | TURN/   |
                 | Proxy   |              | STUN    |
                 |         |              |  Server |
                 +---------+              +---------+


      ............................................................


                                  +----------+
                                  |   /  \   |
            +---------+              /SIP \          +----------+
            | +---------+           /Phone \         |   /  \   |
            | | +---------+        /        \           /SIP \
            | | |         |       ------------         /Phone \
            +-| |   PC    |                           /        \
              +-|         |                          ------------
                +---------+


                                 Enterprise
```

Figure 14: Enterprise Configuration

In this scenario, shown in Figure 14 there is an enterprise that wishes to deploy VoIP. The enterprise has a single site, and there is a firewall/NAT at the border to the public Internet. This NAT is symmetric. Internally, the enterprise is using 10.0.0.0/16. Behind the firewall, within the DMZ, is a TURN/STUN server and a SIP proxy. The firewall has been configured to allow incoming traffic to port 5060 to go to the SIP proxy. It has also allowed incoming UDP traffic on a specific port range to go to the TURN/STUN server. The TURN server has an internal address of 10.0.1.10. This port range contains enough addresses to allow simultaneous conversations to cover the needs of the enterprise, but no more. That range is configured on the TURN/STUN server, so that the TURN server allocates addresses within this range. The TURN server is also configured to allocate two addresses for each allocation request, using the MORE-AVAILABLE feature in TURN [14]. Thats because different addresses need to be used if the TURN server is contacted externally (192.0.2.1) vs. internally (10.0.1.10).

Within the enterprise, PCs and hardphones are used for VoIP. All of them are configured to use the proxy and TURN/STUN server that is run by the enterprise.

All call flows in this section only indicate RTP. The flows for RTCP are not shown.

### 9.2.1 Intra-Enterprise Call

```
             A                 STUN+TURN Server
             |(1) STUN Bind           |
             |s=10.0.1.1:1010         |
             |d=10.0.1.10:3478        |
             |----------------------->|
             |(2) STUN Resp           |
             |s=10.0.1.10:3478        |
             |d=10.0.1.1:1010         |
             |M=10.0.1.1:1010         |
             |<-----------------------|
             |(3) TURN Alloc          |
             |s=10.0.1.1:1010         |
             |d=10.0.1.10:5556        |
             |----------------------->|
             |(4) TURN Resp           |
             |s=10.0.1.10:5556        |
             |d=10.0.1.1:1010         |
             |M=192.0.2.1:8076        |
             |M=10.0.1.10:8076        |
             |<-----------------------|
```

Figure 15: A's Unilateral Allocations

TODO: The flows here don't align yet with the MORE-AVAILABLE mechanism in TURN. Need to update to do so.

The first case to consider is that where a user within the enterprise calls another user within the same enterprise. First, user A performs its unilateral allocations. This is shown in Figure 15. The STUN allocation does not yield a new address, but the TURN allocation, of course, does. In fact, the TURN allocation yields two addresses. The first of these, 192.0.2.1, has a higher priority. As a result, the offer from A to B has three addresses, as shown in Figure 16.

```
   v=0
   o=alice 2890844730 2890844731 IN IP4 host.example.com
   s=
   c=IN IP4 192.0.2.1
   t=0 0
   m=audio 8076 RTP/AVP 0
   a=alt:1 1.0 : user 9kksj== 10.0.1.1 1010
   a=alt:2 0.5 : user1 9kksk== 192.0.2.1 8076
   a=alt:3 0.4 : user2 9kksl== 10.0.1.10 8076
```

Figure 16: A's Offer

B receives this offer. It performs its own unilateral allocations, shown in Figure 17.

```
             B                 STUN+TURN Server
             |(1) STUN Bind            |
             |s=10.0.1.2:23766         |
             |d=10.0.1.10:3478         |
             |------------------------>|
             |(2) STUN Resp            |
             |s=10.0.1.10:3478         |
             |d=10.0.1.2:23766         |
             |M=10.0.1.2:23766         |
             |<------------------------|
             |(3) TURN Alloc           |
             |s=10.0.1.2:23766         |
             |d=10.0.1.10:5556         |
             |------------------------>|
             |(4) TURN Resp            |
             |s=10.0.1.10:5556         |
             |d=10.0.1.2:23766         |
             |M=192.0.2.1:8078         |
             |M=10.0.1.10:8078         |
             |<------------------------|
```

Figure 17: B's Unilateral Allocations

The STUN derived transport address equals its local transport address, so no additional addresses are obtained through that route. TURN provided B with two new addresses. Next, B performs connectivity checks against the three alternatives provided by A. These checks are shown in Figure 18. The connectivity check to the alternate with ID 1, A's local transport address, succeeds, since both users are within the same address realm. The connectivity to check to the alternate with ID 2, which is the TURN server address on the public Internet, fails. This is because the NAT does not support receipt of requests

from internal hosts that are targeted towards internal bindings. As a result, the STUN request is dropped by the NAT. The connectivity check through the TURN relay using its private address succeeds, however, and yields B a new peer derived transport address - 10.0.1.10:5566.

```
             A    TURN + STUN Server     B            NAT
             |(1) STUN Bind|             |             |
             |s=10.0.1.2:23766           |             |
             |d=10.0.1.1:1010            |             |
             |<--------------------------|             |
             |             |             |             |
             |(2) STUN Reply             |             |
             |s=10.0.1.1:1010            |             |
             |d=10.0.1.2:23766           |             |
             |M=10.0.1.2:23766           |             |
             |-------------------------->|             |
             |             |             |             |
             |             |             |             |
             |             |             |(3) STUN Bind|
             |             |             |s=10.0.1.2:23766
             |             |             |d=192.0.2.1:8076
             |             |             |------------>|
             |             |             |             |
             |             |             |Dropped by NAT
             |             |             |             |
             |             |             |             |
             |             |(4) STUN Bind|             |
             |             |s=10.0.1.2:23766           |
             |             |d=10.0.1.10:8076           |
             |             |<------------|             |
             |             |             |             |
             |             |             |             |
             |(5) STUN Bind|             |             |
             |s=10.0.1.10:5556           |             |
             |d=10.0.1.1:1010            |             |
             |<------------|             |             |
             |             |             |             |
             |(6) STUN Reply             |             |
             |s=10.0.1.1:1010            |             |
             |d=10.0.1.10:5556           |             |
             |M=10.0.1.10:5566           |             |
             |------------>|             |             |
             |             |             |             |
             |             |(7) STUN Reply             |
             |             |s=10.0.1.10:8076           |
             |             |d=10.0.1.2:23766           |
```

```
             |             |M=10.0.1.10:5566          |
             |             |------------>|            |
             |             |             |            |
             |             |             |            |
             |             |             |            |
             |             |             |            |
             |             |             |            |
             |             |             |            |
```

Figure 18: B's Connectivity Test

Based on this, B generates the answer shown in Figure 19. Since B has established connectivity to A's local transport address, it begins sending media there.

```
   v=0
   o=bob 2890844730 289084871 IN IP4 host2.example.com
   s=
   c=IN IP4 192.0.2.1
   t=0 0
   m=audio 8078 RTP/AVP 0
   a=alt:4 1.0 : peer as88jl 10.0.1.2 23766
   a=alt:5 0.5 : peer1 as88kl 192.0.2.1 8078
   a=alt:6 0.4 : peer2 as88ll 10.0.1.10 8078
   a=alt:7 0.4 2 peer3 as88ml 10.0.1.10 5556
```

Figure 19: B's Answer

Now, A performs its connectivity checks, shown in Figure 20. These checks indicate that connectivity is established to B's local transport address (10.0.1.2:23766), B's TURN derived transport address (10.0.1.10:8078) and B's peer derived transport address (10.0.1.10:5556). The highest priority address is the local transport address, and so A sends media there. A also discovered a new peer derived transport address, 10.0.1.10:5556. However, it doesn't bother sending it in a new offer, since B connected using a higher priority address. In fact, both A and B can now free their TURN derived addresses. The call proceeds with media flowing directly between A and B, as desired.

```
             A        TURN + STUN Server         B                NAT
             |(1) STUN Bind    |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=10.0.1.2:23766 |                 |                 |
             |---------------------------------->|                 |
             |(2) STUN Reply   |                 |                 |
```

```
             |s=10.0.1.2:23766 |                 |                 |
             |d=10.0.1.1:1010  |                 |                 |
             |M=10.0.1.1:1010  |                 |                 |
             |<----------------------------------|                 |
             |(3) STUN Bind    |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=192.0.2.1:8078 |                 |                 |
             |---------------------------------------------------->|
             |                 |                 |Dropped by NAT   |
             |(4) STUN Bind    |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=10.0.1.10:8078 |                 |                 |
             |---------------->|                 |                 |
             |                 |(5) STUN Bind    |                 |
             |                 |s=10.0.1.10:5556 |                 |
             |                 |d=10.0.1.2:23766 |                 |
             |                 |---------------->|                 |
             |                 |(6) STUN Reply   |                 |
             |                 |s=10.0.1.2:23766 |                 |
             |                 |d=10.0.1.10:5556 |                 |
             |                 |M=10.0.1.10:5556 |                 |
             |                 |<----------------|                 |
             |(7) STUN Reply   |                 |                 |
             |s=10.0.1.10:8078 |                 |                 |
             |d=10.0.1.1:1010  |                 |                 |
             |M=10.0.1.10:5556 |                 |                 |
             |<----------------|                 |                 |
             |(8) STUN Bind    |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=10.0.1.10:5556 |                 |                 |
             |---------------->|                 |                 |
             |                 |(9) STUN Bind    |                 |
             |                 |s=10.0.1.10:8076 |                 |
             |                 |d=10.0.1.2:23766 |                 |
             |                 |---------------->|                 |
             |                 |(10) STUN Reply  |                 |
             |                 |s=10.0.1.2:23766 |                 |
             |                 |d=10.0.1.10:8076 |                 |
             |                 |M=10.0.1.10:8076 |                 |
             |                 |<----------------|                 |
             |(11) STUN Reply  |                 |                 |
             |s=10.0.1.10:5556 |                 |                 |
             |d=10.0.1.1:1010  |                 |                 |
             |M=10.0.1.10:8076 |                 |                 |
             |<----------------|                 |                 |
```

Figure 20: A's Connectivity Checks

### 9.2.2 Extra-Enterprise Call

In this case, user A within the enterprise calls some user B, not within the enterprise. B is connected to the Internet through a PSTN gateway, and as a result, appears as a UA on the public Internet. Presumably this is some gateway run by a third party termination provider that is being used by the enterprise. Furthermore, this gateway does not support ICE at all, and so will ignore the alt parameters in the SDP.

First, A performs its unilateral allocations. This proceeds identically as shown in Figure 15. It generates the same offer as shown in Figure 16. This gets routed to the called party on the public Internet. This party generates an answer. However, since the called party does not support ICE, the answer has no alt attributes. It has a single IP address and port listed in the c and m lines. As a result, the caller, A, sends media there. Furthermore the called party uses the IP address and port in the m and c lines of A's offer. This is 192.0.2.1:8076, which routes to the enterprise NAT, and is then forwarded to the TURN server, and from there, relayed to the caller. As a result, media now flows in both directions.

OPEN ISSUE: This assumes that the firewall policy admits outbound UDP packets towards any destination. It is likely that some enterprises will not permit this. To fix this, it would require the ability of a client to send media using a TURN relay, similar to the way instant message senders can use a relay in the Message Session Relay Protocol (MSRP) [19].

### 9.2.3 Disconnected Intra-Enterprise

In this scenario, A and B are both within the enterprise. However, they work in different departments. Both departments are connected to the company backbone. However, A's department has a symmetric NAT between it and the company backbone, with user A on the private side. A's department NATs into 10.0.2.0/24. That is, it has allocated 256 of the company's addresses to NAT into. B's department has decided to use 192.168.0.0/16 in its private network.

```
             A                     Dept NAT            STUN+TURN Server
             |(1) STUN Bind            |                         |
             |s=192.168.1.1:1010       |                         |
             |d=10.0.1.10:3478         |                         |
             |------------------------>|                         |
             |                         |(2) STUN Bind            |
             |                         |s=10.0.2.88:6584         |
```

```
             |                        |d=10.0.1.10:3478        |
             |                        |----------------------->|
             |                        |(3) STUN Resp           |
             |                        |s=10.0.1.10:3478        |
             |                        |d=10.0.2.88:6584        |
             |                        |M=10.0.2.88:6584        |
             |                        |<-----------------------|
             |(4) STUN Resp           |                        |
             |s=10.0.1.10:3478        |                        |
             |d=192.168.1.1:1010      |                        |
             |M=10.0.2.88:6584        |                        |
             |<-----------------------|                        |
             |(5) TURN Alloc          |                        |
             |s=192.168.1.1:1010      |                        |
             |d=10.0.1.10:5556        |                        |
             |----------------------->|                        |
             |                        |(6) TURN Alloc          |
             |                        |s=10.0.2.88:6585        |
             |                        |d=10.0.1.10:5556        |
             |                        |----------------------->|
             |                        |(7) TURN Alloc          |
             |                        |s=10.0.1.10:5556        |
             |                        |d=10.0.2.88:6585        |
             |                        |M=192.2.0.1:8076        |
             |                        |M=10.0.1.10:8076        |
             |                        |<-----------------------|
             |(8) TURN Alloc          |                        |
             |s=10.0.1.10:5556        |                        |
             |d=192.168.1.1:1010      |                        |
             |M=192.2.0.1:8076        |                        |
             |M=10.0.1.10:8076        |                        |
             |<-----------------------|                        |
```

Figure 21: A's Unilateral Allocations

A starts by performing its usual unilateral allocations, as shown in Figure 21. These allocations provide it a STUN derived and two TURN derived transport address. Based on these, it sends the offer shown in Figure 22.

```
   v=0
   o=alice 2890844730 2890844731 IN IP4 host.example.com
   s=
   c=IN IP4 192.0.2.1
   t=0 0
   m=audio 8076 RTP/AVP 0
   a=alt:1 1.0 : user 9kksj== 192.168.1.1 1010
   a=alt:2 0.8 : user5 sad87 10.0.2.88:6584
   a=alt:3 0.5 : user1 9kksk== 192.0.2.1 8076
   a=alt:4 0.4 : user2 9kksl== 10.0.1.10 8076
```

Figure 22: A's Offer

B performs its unilateral allocations. (Figure 17. Then it performs its connectivity checks. This is where things differ. B's checks are shown in Figure 23. The STUN request to A's local transport address, 192.168.1.1:1010 fails (message 1), because 192.168.0.0/16 is not reachable from B. The STUN request (message 2) to A's STUN derived transport address, 10.0.2.88:6584, also fails, because the department NAT is symmetric (it would have worked if it was full-cone). The STUN request to A's public TURN derived transport address, 192.0.2.1:8076, also fails since the NAT won't "turn around" such packets. The final attempt, B's STUN request to A's private TURN derived transport address, does in fact succeed. It yields a new peer derived transport address for B, 10.0.1.10:5566.

```
             A         Dept NAT TURN + STUN Server     B         Corp. NAT
             |             |             |             |(1) STUN Bind|
             |             |             |             |s=10.0.1.2:23766
             |             |             |             |d=192.168.1.1:1010
             |             |             |             |------------>|
             |             |             |             |Dropped      |
             |             |(2) STUN Bind|             |             |
             |             |s=10.0.1.2:23766           |             |
             |             |d=10.0.2.88:6584           |             |
             |             |<--------------------------|             |
             |             |Dropped      |             |             |
             |             |             |             |(3) STUN Bind|
             |             |             |             |s=10.0.1.2:23766
             |             |             |             |d=192.0.2.1:8076
             |             |             |             |------------>|
             |             |             |             |Dropped by NAT
             |             |             |(4) STUN Bind|             |
             |             |             |s=10.0.1.2:23766           |
             |             |             |d=10.0.1.10:8076           |
             |             |             |<------------|             |
             |             |(5) STUN Bind|             |             |
```

```
             |             |s=10.0.1.10:5556           |             |
             |             |d=10.0.2.88:6585           |             |
             |             |<------------|             |             |
             |(6) STUN Bind|             |             |             |
             |s=10.0.1.10:5556           |             |             |
             |d=192.168.1.1:1010         |             |             |
             |<------------|             |             |             |
             |(7) STUN Reply             |             |             |
             |s=192.168.1.1:1010         |             |             |
             |d=10.0.1.10:5556           |             |             |
             |M=10.0.1.10:5566           |             |             |
             |------------>|             |             |             |
             |             |(8) STUN Reply             |             |
             |             |s=10.0.2.88:6585           |             |
             |             |d=10.0.1.10:5556           |             |
             |             |M=10.0.1.10:5566           |             |
             |             |------------>|             |             |
             |             |             |(9) STUN Reply             |
             |             |             |s=10.0.1.10:8076           |
             |             |             |d=10.0.1.2:23766           |
             |             |             |M=10.0.1.10:5566           |
             |             |             |------------>|             |
```

Figure 23: B's Connectivity Checks

Based on this, B generates the answer shown in Figure 24.

```
v=0
o=bob 2890844730 289084871 IN IP4 host2.example.com
s=
c=IN IP4 192.0.2.1
t=0 0
m=audio 8078 RTP/AVP 0
a=alt:5 1.0 : peer as88jl 10.0.1.2 23766
a=alt:6 0.5 : peer1 as88kl 192.0.2.1 8078
a=alt:7 0.4 : peer2 as88ll 10.0.1.10 8078
a=alt:8 0.4 4 peer3 as88ml 10.0.1.10 5556
```

Figure 24: B's Answer

B receives this answer. It performs its connectivity checks against the four addresses provided by B. These checks are shown in Figure 25. The first check, to B's local transport address (10.0.1.2:23766) actually succeeds. Indeed, it also provides A with a new peer derived transport address - 10.0.2.88:6586. The second check, to B's TURN derived transport address on the public Internet, fails, because the NAT won't turn around packets. The third check, to B's TURN derived

   transport address on the private corporate network, succeeds as
   expected, and provides A with a new peer derived transport address,
   10.0.1.10:5556. The fourth check, to B's peer derived transport
   address (through the TURN relay), also succeeds.

```
             A            Dept NAT    TURN + STUN Server        B        Corp.
NAT
             |(1) STUN Bind   |                 |
|               |
             |s=192.168.1.1:1010                |
|               |
             |d=10.0.1.2:23766|                 |
|               |
             |--------------->|                 |
|               |
             |                |(2) STUN Bind    |
|               |
             |                |s=10.0.2.88:6586|
|               |
             |                |d=10.0.1.2:23766|
|               |
             |                |--------------------------------
>|               |
             |                |(3) STUN Reply  |
|               |
             |                |s=10.0.1.2:23766|
|               |
             |                |d=10.0.2.88:6586|
|               |
             |                |M=10.0.2.88:6586|
|               |
             |                |
<--------------------------------|                |
             |(4) STUN Reply  |                 |
|               |
             |s=10.0.1.2:23766|                 |
|               |
             |d=192.168.1.1:1010                |
|               |
             |M=10.0.2.88:6586|                 |
|               |
             |<---------------|                 |
|               |
             |(5) STUN Bind   |                 |
|               |
             |s=192.168.1.1:1010                |
|               |
             |d=192.0.2.1:8078|                |
|                |

|------------------------------------------------------------------->|
             |                |                |                |Dropped by
NAT  |
             |(6) STUN Bind   |                |
|                |
             |s=192.168.1.1:1010               |
|                |
             |d=10.0.1.10:8078|                |
|                |
             |--------------->|                |
|                |
             |                |(7) STUN Bind   |
|                |
             |                |s=10.0.2.88:6587|
|                |
             |                |d=10.0.1.10:8078|
|                |
             |                |--------------->|
|                |
             |                |                |(8) STUN Bind
|                |
             |                |                |
s=10.0.1.10:5556|                |
             |                |                |
d=10.0.1.2:23766|                |
             |                |                |---------------
>|                |
             |                |                |(9) STUN Reply
|                |
             |                |                |
s=10.0.1.2:23766|                |
             |                |                |
d=10.0.1.10:5556|                |
             |                |                |
M=10.0.1.10:5556|                |
             |                |                |
<---------------|                |
             |                |(10) STUN Reply |
|                |
             |                |s=10.0.1.10:8078|
|               |
             |                |d=10.0.2.88:6587|
|               |
             |                |M=10.0.1.10:5556|
|               |
             |                |<---------------|
|               |
             |(11) STUN Reply |                |
|               |
             |s=10.0.1.10:8078|                |
|               |
             |d=192.168.1.1:1010               |
|               |
             |M=10.0.1.10:5556|                |
|               |
             |<---------------|                |
|               |
             |(12) STUN Bind  |                |
|               |
             |s=192.168.1.1:1010               |
|               |
             |d=10.0.1.10:5556|                |
|               |
             |--------------->|                |
|               |
             |                |(13) STUN Bind  |
|               |
             |                |s=10.0.2.88:6585|
|               |
             |                |d=10.0.1.10:5556|
|               |
             |                |--------------->|
|               |
             |                |                |(14) STUN Bind
|               |
             |                |                |
s=10.0.1.10:8076|                |
             |                |                |
d=10.0.1.2:23766|                |
             |                |                |---------------
>|               |
             |                |                |(15) STUN Reply
|               |
             |                |                |
s=10.0.1.2:23766|                |
             |                |                |
d=10.0.1.10:8076|                |
             |                |                |
M=10.0.1.10:8076|                |
             |                |                |
<---------------|                |
             |                |(16) STUN Reply |
|                |
             |                |s=10.0.1.10:5556|
|                |
             |                |d=10.0.2.88:6585|
|                |
             |                |M=10.0.1.10:8076|
|                |
             |                |<---------------|
|                |
             |(17) STUN Reply |                |
|                |
             |s=10.0.1.10:5556|                |
|                |
             |d=192.168.1.1:1010              |
|                |
             |M=10.0.1.10:8076|                |
|                |
             |<---------------|                |
|                |
```

Figure 25: A's Connectivity Checks

At this point, the highest priority address that A has established connectivity to is B's local transport address. So, A begins sending media there. However, the connectivity checks provided A with two new addresses. One of them is of higher priority than the highest priority address that B has connected to. So, using a re-INVITE or an UPDATE [6], A will generate a new offer to B:

```
   v=0
   o=alice 2890844730 2890844732 IN IP4 host.example.com
   s=
   c=IN IP4 192.0.2.1
   t=0 0
   m=audio 8076 RTP/AVP 0
   a=alt:1 1.0 : user 9kksj== 192.168.1.1 1010
   a=alt:2 0.8 : user5 sad87 10.0.2.88:6584
   a=alt:3 0.5 : user1 9kksk== 192.0.2.1 8076
   a=alt:4 0.4 : user2 9kksl== 10.0.1.10 8076
   a=alt:5 1.0 5 user6 77==8ja 10.0.2.88 6586
   a=alt:6 0.4 7 user7 a8kkzu 10.0.1.10 5556
```

Figure 26: A's 2nd Offer

When B receives this offer, it sees two new alternates, 5 and 6. So, it performs connectivity checks to them, as shown in Figure 27. The first check succeeds, and tells B that its address as seen by A is 10.0.1.2:23766. This is not a new address. The second check also succeeds, but doesn't provide B with a new address. However, B now has a higher priority address with connectivity to A. It therefore begins sending media there. B will generate an answer, of course, but it is identical to its previous answer, in terms of addresses.

```
             A         Dept NAT TURN + STUN Server     B         Corp. NAT
             |             |(1) STUN Bind|             |             |
             |             |s=10.0.1.2:23766           |             |
             |             |d=10.0.2.88:6586           |             |
             |             |<--------------------------|             |
             |(2) STUN Bind|             |             |             |
             |s=10.0.1.2:23766           |             |             |
             |d=192.168.1.1:1010         |             |             |
             |<------------|             |             |             |
             |(3) STUN Reply             |             |             |
             |s=192.168.1.1:1010         |             |             |
             |d=10.0.1.2:23766           |             |             |
             |M=10.0.1.2:23766           |             |             |
             |------------>|             |             |             |
             |             |(4) STUN Reply             |             |
             |             |s=10.0.2.88:6586           |             |
             |             |d=10.0.1.2:23766           |             |
             |             |M=10.0.1.2:23766           |             |
             |             |-------------------------->|             |
             |             |             |(5) STUN Bind|             |
             |             |             |s=10.0.1.2:23766           |
             |             |             |d=10.0.1.10:5556           |
             |             |             |<------------|             |
```

```
             |             |(6) STUN Bind|             |             |
             |             |s=10.0.1.10:8078           |             |
             |             |d=10.0.2.88:6585           |             |
             |             |<------------|             |             |
             |(7) STUN Bind|             |             |             |
             |s=10.0.1.10:8078           |             |             |
             |d=192.168.1.1:1010         |             |             |
             |<------------|             |             |             |
             |(8) STUN Reply             |             |             |
             |s=192.168.1.1:1010         |             |             |
             |d=10.0.1.10:8078           |             |             |
             |M=10.0.1.10:8078           |             |             |
             |------------>|             |             |             |
             |             |(9) STUN Reply             |             |
             |             |s=10.0.2.88:6585           |             |
             |             |d=10.0.1.10:8078           |             |
             |             |M=10.0.1.10:8078           |             |
             |             |------------>|             |             |
             |             |             |(10) STUN Reply            |
             |             |             |s=10.0.1.10:5556           |
             |             |             |d=10.0.1.2:23766           |
             |             |             |M=10.0.1.10:8078           |
             |             |             |------------>|             |
```

Figure 27: B's 2nd Connectivity Checks

The net result of this scenario is that, after two ICE cycles, A and B are able to send media to each other directly, even though there is a symmetric NAT between them.

## 9.3 Centrex

In a centrex scenario, a third party provider owns and operates the SIP and TURN/STUN servers. The enterprise merely changes their firewall configuration to allow SIP traffic out to port 5060 to the provider's SIP proxy, and to allow TURN traffic out to port 5556 and 3478, on the provider's TURN/STUN server. The corporate NAT is symmetric. The TURN/STUN server runs on 192.0.2.10. This scenario is shown in Figure 28.

```
                              Provider Equipment

                         +---------+   +---------+
                         |         |   | TURN/   |
                         | Proxy   |   | STUN    |
                         |         |   |  Server |
                         +---------+   +---------+
```

```
                                                      Public
                                                      Internet



                                    192.0.2.1
                                 +---------+
                                 |         |
           ----------------------| Firewall|--------------------------
                                 |  NAT    |
            10.0.0.0/16          +---------+



                                  +----------+
                                  |   /  \   |
            +---------+              /SIP \          +----------+
            | +---------+           /Phone \         |   /  \   |
            | | +---------+        /        \           /SIP \
            | | |         |       ------------         /Phone \
            +-| |   PC    |                           /        \
              +-|         |                          ------------
                +---------+


                                 Enterprise
```

Figure 28: Centrex Configuration

## 9.3.1 Intra-Enterprise Call

In this scenario, user A calls user B. Both are within the enterprise. First, A performs its unilateral allocations. These are shown in Figure 29. These yield a STUN derived transport address and a TURN derived transport address. A sends these in the offer shown in Figure 30.

```
             A                    Corp. NAT            STUN+TURN Server
             |(1) STUN Bind           |                        |
             |s=10.0.1.1:1010         |                        |
             |d=192.0.2.10:3478       |                        |
             |----------------------->|                        |
             |                        |(2) STUN Bind           |
             |                        |s=192.0.2.1:9988        |
             |                        |d=192.0.2.10:3478       |
             |                        |----------------------->|
```

```
             |                        |(3) STUN Resp           |
             |                        |s=192.0.2.10:3478       |
             |                        |d=192.0.2.1:9988        |
             |                        |M=192.0.2.1:9988        |
             |                        |<-----------------------|
             |(4) STUN Resp           |                        |
             |s=192.0.2.10:3478       |                        |
             |d=10.0.1.1:1010         |                        |
             |M=192.0.2.1:9988        |                        |
             |<-----------------------|                        |
             |(5) TURN Alloc          |                        |
             |s=10.0.1.1:1010         |                        |
             |d=192.0.2.10:5556       |                        |
             |----------------------->|                        |
             |                        |(6) TURN Alloc          |
             |                        |s=192.0.2.1:9989        |
             |                        |d=192.0.2.10:5556       |
             |                        |----------------------->|
             |                        |(7) TURN Resp           |
             |                        |s=192.0.2.10:5556       |
             |                        |d=192.0.1.1:9989        |
             |                        |M=192.0.2.10:8076       |
             |                        |<-----------------------|
             |(8) TURN Resp           |                        |
             |s=192.0.2.10:5556       |                        |
             |d=10.0.1.1:1010         |                        |
             |M=192.0.2.10:8076       |                        |
             |<-----------------------|                        |
```

Figure 29: A's Unilateral Allocations

```
   v=0
   o=alice 2890844730 2890844731 IN IP4 host.example.com
   s=
   c=IN IP4 192.0.2.10
   t=0 0
   m=audio 8076 RTP/AVP 0
   a=alt:1 1.0 : user 9kksj== 10.0.1.1 1010
   a=alt:2 0.5 : user1 9kksk== 192.0.2.1 9988
   a=alt:3 0.4 : user2 9kksl== 192.0.2.10 8076
```

Figure 30: A's Offer

This offer is received by B. B performs its unilateral allocations, shown in Figure 31. These yield a STUN derived and TURN derived transport address.

```
             B                    Corp. NAT            STUN+TURN Server
             |(1) STUN Bind            |                         |
             |s=10.0.1.2:23766         |                         |
             |d=192.0.2.10:3478        |                         |
             |------------------------>|                         |
             |                         |(2) STUN Bind            |
             |                         |s=192.0.2.1:9990         |
             |                         |d=192.0.2.10:3478        |
             |                         |------------------------>|
             |                         |(3) STUN Resp            |
             |                         |s=192.0.2.10:3478        |
             |                         |d=192.0.2.1:9990         |
             |                         |M=192.0.2.1:9990         |
             |                         |<------------------------|
             |(4) STUN Resp            |                         |
             |s=192.0.2.10:3478        |                         |
             |d=10.0.1.2:23766         |                         |
             |M=192.0.2.1:9990         |                         |
             |<------------------------|                         |
             |(5) TURN Alloc           |                         |
             |s=10.0.1.2:23766         |                         |
             |d=192.0.2.10:5556        |                         |
             |------------------------>|                         |
             |                         |(6) TURN Alloc           |
             |                         |s=192.0.2.1:9991         |
             |                         |d=192.0.2.10:5556        |
             |                         |------------------------>|
             |                         |(7) TURN Resp            |
             |                         |s=192.0.2.10:5556        |
             |                         |d=192.0.2.1:9991         |
             |                         |M=192.0.2.10:8078        |
```

```
             |                        |<-----------------------|
             |(8) TURN Resp           |                        |
             |s=192.0.2.10:5556       |                        |
             |d=10.0.1.2:23766        |                        |
             |M=192.0.2.10:8078       |                        |
             |<-----------------------|                        |
```

Figure 31: B's Unilateral Allocations

Now, B begins its connectivity checks, as shown in Figure 32. The first check (message 1), to A's local transport address, 10.0.1.1:1010, succeeds, since A and B are behind the same NAT. The second check, to A's STUN derived transport address, fails, since the enterprise NAT won't turn around packets. The third check, to A's TURN derived transport address, 192.0.2.10:8076, also succeeds, and yields B a new peer derived transport address, 192.0.2.10:5556.

```
             A                B            Corp. NAT   TURN + STUN Server
             |(1) STUN Bind   |                |                |
             |s=10.0.1.2:23766|                |                |
             |d=10.0.1.1:1010 |                |                |
             |<---------------|                |                |
             |(2) STUN Reply  |                |                |
             |s=10.0.1.1:1010 |                |                |
             |d=10.0.1.2:23766|                |                |
             |M=10.0.1.2:23766|                |                |
             |--------------->|                |                |
             |                |(3) STUN Bind   |                |
             |                |s=10.0.1.2:23766|                |
             |                |d=192.0.2.1:9988|                |
             |                |--------------->|                |
             |                |                |Dropped         |
             |                |(4) STUN Bind   |                |
             |                |s=10.0.1.2:23766|                |
             |                |d=192.0.2.10:8076                |
             |                |--------------->|                |
             |                |                |(5) STUN Bind   |
             |                |                |s=192.0.2.1:9992|
             |                |                |d=192.0.2.10:8076
             |                |                |--------------->|
             |                |                |(6) STUN Bind   |
             |                |                |s=192.0.2.10:5556
             |                |                |d=192.0.2.1:9988|
             |                |                |<---------------|
             |(7) STUN Bind   |                |                |
             |s=192.0.2.10:5556                |                |
             |d=10.0.1.1:1010 |                |                |
```

```
             |<--------------------------------|                |
             |(8) STUN Reply  |                |                |
             |s=10.0.1.1:1010 |                |                |
             |d=192.0.2.10:5556                |                |
             |M=192.0.2.10:5556                |                |
             |------------------------------->|                |
             |                |                |(9) STUN Reply  |
             |                |                |s=192.0.2.1:9988|
             |                |                |d=192.0.2.10:5556
             |                |                |M=192.0.2.10:5556
             |                |                |--------------->|
             |                |                |(10) STUN Reply |
             |                |                |s=192.0.2.10:8076
             |                |                |d=192.0.2.1:9992|
             |                |                |M=192.0.2.10:5556
             |                |                |<---------------|
             |                |(11) STUN Reply |                |
             |                |s=192.0.2.10:8076                |
             |                |d=10.0.1.2:23766|                |
             |                |M=192.0.2.10:5556                |
             |                |<---------------|                |
```

Figure 32: B's Connectivity Checks

B can now send media to A directly. It also generates an answer, shown in Figure 33.

```
v=0
o=bob 2890844730 289084871 IN IP4 host2.example.com
s=
c=IN IP4 192.0.2.10
t=0 0
m=audio 8078 RTP/AVP 0
a=alt:4 1.0 : peer as88jl 10.0.1.2 23766
a=alt:5 0.8 : peer1 as88kl 192.0.2.1 9990
a=alt:6 0.4 : peer2 as88ll 192.0.2.10 8078
a=alt:7 0.4 : peer3 as88ml 192.0.2.10 5556
```

Figure 33: B's Answer

This arrives at A. A is able to send media immediately to B using the default, 192.0.2.10:8078. It also starts its connectivity checks to find a better choice. These checks are shown in Figure 34. As expected, the check for connectivity to 10.0.1.2:23766 succeeds, representing the highest priority address. The check to 192.0.2.1:9990 fails, because the NAT won't turn around internal packets. The checks to 192.0.2.10:8078 and 192.0.2.10:5556 succeed,

   and the former resuls in a peer-derived transport address of
   192.0.2.10:5556. However, A knows that B has already connected to a
   higher priority address, so it doesn't bother with an additional
   offer/answer exchange.

```
             A                  B             Corp. NAT    TURN + STUN Server
             |(1) STUN Bind    |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=10.0.1.2:23766 |                 |                 |
             |---------------->|                 |                 |
             |(2) STUN Reply   |                 |                 |
             |s=10.0.1.2:23766 |                 |                 |
             |d=10.0.1.1:1010  |                 |                 |
             |M=10.0.1.1:1010  |                 |                 |
             |<----------------|                 |                 |
             |(3) STUN Bind    |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=192.0.2.1:9990 |                 |                 |
             |---------------------------------->|                 |
             |                 |                 |Dropped          |
             |(4) STUN Bind    |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=192.0.2.10:8078|                 |                 |
             |---------------------------------->|                 |
             |                 |                 |(5) STUN Bind    |
             |                 |                 |s=192.0.2.1:9992 |
             |                 |                 |d=192.0.2.10:8078|
             |                 |                 |---------------->|
             |                 |                 |(6) STUN Bind    |
             |                 |                 |s=192.0.2.10:5556|
             |                 |                 |d=192.0.2.1:9991 |
             |                 |                 |<----------------|
             |                 |(7) STUN Bind    |                 |
             |                 |s=192.0.2.10:5556|                 |
             |                 |d=10.0.1.2:23766 |                 |
             |                 |<----------------|                 |
             |                 |(8) STUN Reply   |                 |
             |                 |s=10.0.1.2:23766 |                 |
             |                 |d=192.0.2.10:5556|                 |
             |                 |M=192.0.2.10:5556|                 |
             |                 |---------------->|                 |
             |                 |                 |(9) STUN Reply   |
             |                 |                 |s=192.0.2.1:9991 |
             |                 |                 |d=192.0.2.10:5556|
             |                 |                 |M=192.0.2.10:5556|
             |                 |                 |---------------->|
             |                 |                 |(10) STUN Reply  |
```

```
             |                 |                 |s=192.0.2.10:8078|
             |                 |                 |d=192.0.2.1:9992 |
             |                 |                 |M=192.0.2.10:5556|
             |                 |                 |<----------------|
             |(11) STUN Reply  |                 |                 |
             |s=192.0.2.10:8078|                 |                 |
             |d=10.0.1.1:1010  |                 |                 |
             |M=192.0.2.10:5556|                 |                 |
             |<----------------------------------|                 |
             |(12) STUN Bind   |                 |                 |
             |s=10.0.1.1:1010  |                 |                 |
             |d=192.0.2.10:5556|                 |                 |
             |---------------------------------->|                 |
             |                 |                 |(13) STUN Bind   |
             |                 |                 |s=192.0.2.1:9989 |
             |                 |                 |d=192.0.2.10:5556|
             |                 |                 |---------------->|
             |                 |                 |(14) STUN Bind   |
             |                 |                 |s=192.0.2.10:8076|
             |                 |                 |d=192.0.2.1:9991 |
             |                 |                 |<----------------|
             |                 |(15) STUN Bind   |                 |
             |                 |s=192.0.2.10:8076|                 |
             |                 |d=10.0.1.2:23766 |                 |
             |                 |<----------------|                 |
             |                 |(16) STUN Reply  |                 |
             |                 |s=10.0.1.2:23766 |                 |
             |                 |d=192.0.2.10:8076|                 |
             |                 |M=192.0.2.10:8076|                 |
             |                 |---------------->|                 |
             |                 |                 |(17) STUN Reply  |
             |                 |                 |s=192.0.2.1:9991 |
             |                 |                 |d=192.0.2.10:8076|
             |                 |                 |M=192.0.2.10:8076|
             |                 |                 |---------------->|
             |                 |                 |(18) STUN Reply  |
             |                 |                 |s=192.0.2.10:5556|
             |                 |                 |d=192.0.2.1:9989 |
             |                 |                 |M=192.0.2.10:8076|
             |                 |                 |<----------------|
             |(19) STUN Reply  |                 |                 |
             |s=192.0.2.10:5556|                 |                 |
             |d=10.0.1.1:1010  |                 |                 |
             |M=192.0.2.10:8076|                 |                 |
             |<----------------------------------|                 |
```

Figure 34: A's Connectivity Checks

The conclusion is that A and B communicate directly, without using the provider's relay. They can proceed to de-allocate the TURN addresses once the call is active.

## 10. Security Considerations

Security considerations are discussed throughout the document.
[[Editors Note: Need to summarize here anyway.]].

## 11. IANA Considerations

This specification registers a new precondition type and a new SDP attributes.

### 11.1 Precondition Type

Name: connectivity

Description: Confirm end-to-end connectivity with STUN.

### 11.2 SDP Attributes

This specification defines one new media attribute: alt. Its syntax is defined in Section 7.

## 12. IAB Considerations

The IAB has studied the problem of "Unilateral Self Address Fixing", which is the general process by which a client attempts to determine its address in another realm on the other side of a NAT through a collaborative protocol reflection mechanism [8]. ICE is an example of a protocol that performs this type of function. Interestingly, the process for ICE is not unilateral, but bilateral, and the difference has a signficant impact on the issues raised by IAB. The IAB has mandated that any protocols developed for this purpose document a specific set of considerations. This section meets those requirements.

### 12.1 Problem Definition

From RFC 3424 any UNSAF proposal must provide:

> Precise definition of a specific, limited-scope problem that is to be solved with the UNSAF proposal. A short term fix should not be generalized to solve other problems; this is why "short term fixes usually aren't".

The specific problems being solved by ICE are:

> Provide a means for two peers to determine the set of transport addresses which can be used for communication.
>
> Provide a means for resolving many of the limitations of other UNSAF mechanisms by wrapping them in an additional layer of processing (the ICE methodology).
>
> Provide a means for a client to determine an address that is reachable by another peer with which it wishes to communicate.

### 12.2 Exit Strategy

From RFC 3424, any UNSAF proposal must provide:

> Description of an exit strategy/transition plan. The better short term fixes are the ones that will naturally see less and less use as the appropriate technology is deployed.

ICE itself doesn't easily get phased out. However, it is useful even in a globally connected Internet, to serve as a means for detecting whether a router failure has temporarily disrupted connectivity, for example. However, what ICE does is help phase out other UNSAF mechanisms. ICE effectively selects amongst those mechanisms,

prioritizing ones that are better, and deprioritizing ones that are worse. Local IPv6 addresses are always the most preferred. As NATs begin to dissipate as IPv6 is introduced, derived transport addresses from other UNSAF mechanisms simply never get used, because higher priority connectivity exists. Therefore, the servers get used less and less, and can eventually be remove when their usage goes to zero.

Indeed, ICE can assist in the transition from IPv4 to IPv6. It can be used to determine whether to use IPv6 or IPv4 when two dual-stack hosts communicate with SIP (IPv6 gets used). It can also allow a client in a v6 island to communicate with a v4 host on the other side of a 6to4 NAT, by allowing the v6 host to address-fix against the v4 host, and in the process, obtain a v4 address which can be handed to the v4 client.

## 12.3 Brittleness Introduced by ICE

From RFC3424, any UNSAF proposal must provide:

> Discussion of specific issues that may render systems more "brittle".  For example, approaches that involve using data at multiple network layers create more dependencies, increase debugging challenges, and make it harder to transition.

ICE actually removes brittleness from existing UNSAF mechanisms. In particular, traditional STUN (the usage described in RFC 3489) has several points of brittleness. One of them is the discovery process which requires a client to try and classify the type of NAT it is behind. This process is error-prone. With ICE, that discovery process is simply not used. Rather than unilaterally assessing the validity of the address, its validity is dynamically determined by measuring connectivity to a peer. The process of determining connectivity is very robust. The only potential problem is that bilaterally fixed addresses through STUN can expire if traffic does not keep them alive. However, that is substantially less brittleness than the STUN discovery mechanisms.

Another point of brittleness in STUN, TURN, and any other unilateral mechanism is its absolute reliance on an additional server. ICE makes use of a server for allocating unilateral addresses, but allows clients to directly connect if possible. Therefore, in some cases, the failure of a STUN or TURN server would still allow for a call to progress when ICE is used.

Another point of brittleness in traditional STUN is that it assumes that the STUN server is on the public Internet. Interestingly, with ICE, that is not necessary. There can be a multitude of STUN servers in a variety of address realms. ICE will discover the one that has

provided a usable address.

The most troubling point of brittleness in traditional STUN is that it doesn't work in all network topologies. In cases where there is a shared NAT between each client and the STUN server, traditional STUN may not work. With ICE, that restriction can be lifted.

Traditional STUN also introduces some security considerations. Unfortunately, since ICE still uses network resident STUN servers, those security considerations still exist.

### 12.4 Requirements for a Long Term Solution

From RFC 3424, any UNSAF proposal must provide:

> Identify requirements for longer term, sound technical solutions -- contribute to the process of finding the right longer term solution.

Our conclusions from STUN remain unchanged. However, we feel ICE actually helps because we believe it can be part of the long term solution.

### 12.5 Issues with Existing NAPT Boxes

From RFC 3424, any UNSAF proposal must provide:

> Discussion of the impact of the noted practical issues with existing, deployed NA[P]Ts and experience reports.

A number of NAT boxes are now being deployed into the market which try and provide "generic" ALG functionality. These generic ALGs hunt for IP addresses,  either in text or binary form within a packet, and rewrite them if they match a binding. This will interfere with proper operation of any UNSAF mechanism, including ICE.

## 13. Acknowledgements

The authors would like to thank Douglas Otis and Francois Audet for their comments and input.

## Informative References

[1] 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.

[2] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A. and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[3] Borella, M., Lo, J., Grabelsky, D. and G. Montenegro, "Realm Specific IP: Framework", RFC 3102, October 2001.

[4] Borella, M., Grabelsky, D., Lo, J. and K. Taniguchi, "Realm Specific IP: Protocol Specification", RFC 3103, October 2001.

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

[6] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

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

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

[9] Camarillo, G., Eriksson, G., Holler, J. and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

[10] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[11] Rosenberg, J., Schulzrinne, H. and J. Weinberger, "An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing", draft-ietf-sip-symmetric-response-00 (work in progress), September 2002.

[12] Mahy, R., "Requirements for Connection Reuse in the Session Initiation Protocol (SIP)", draft-ietf-sipping-connect-reuse-reqs-00 (work in progress), October 2002.

[13] Rosenberg, J., Weinberger, J., Huitema, C. and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[14] Rosenberg, J., "Traversal Using Relay NAT (TURN)", draft-rosenberg-midcom-turn-01 (work in progress), March 2003.

[15] Shore, M., "The TIST (Topology-Insensitive Service Traversal) Protocol", draft-shore-tist-prot-00 (work in progress), May 2002.

[16] Yon, D., "Connection-Oriented Media Transport in SDP", draft-ietf-mmusic-sdp-comedia-05 (work in progress), March 2003.

[17] Huitema, C., "RTCP attribute in SDP", draft-ietf-mmusic-sdp4nat-05 (work in progress), June 2003.

[18] Rosenberg, J., Mahy, R. and S. Sen, "NAT and Firewall Scenarios and Solutions for SIP", draft-ietf-sipping-nat-scenarios-00 (work in progress), June 2002.

[19] Campbell, B., "Instant Message Sessions in SIMPLE", draft-ietf-simple-message-sessions-00 (work in progress), May 2003.

[20] Camarillo, G. and J. Rosenberg, "The Alternative Semantics for the Session Description Protocol Grouping Framework", draft-camarillo-mmusic-alt-01 (work in progress), June 2003.

[21] Audet, F., Aoun, C., Sen, S. and F. Meijer, "Identifying Intra-realm Calls using STUN", draft-sen-sipping-intrarealm-stun-00 (work in progress), September 2002.

[22] Huitema, C., "Teredo: Tunneling IPv6 over UDP through NATs", draft-ietf-ngtrans-shipworm-08 (work in progress), September 2002.

## Author's Address

Jonathan Rosenberg
dynamicsoft
600 Lanidex Plaza
Parsippany, NJ  07054
US

Phone: +1 973 952-5000
EMail: jdrosen@dynamicsoft.com
URI:   http://www.jdrosen.net

## Intellectual Property Statement

## Full Copyright Statement

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.