Electronic Communications Committee (ECC)
within the European Conference of Postal and Telecommunications Administrations (CEPT)

TECHNICAL ISSUES OF ESTABLISHING ANY-TO-ANY 2-WAY REAL-TIME COMMUNICATIONS OVER THE INTERNET

Gothenburg, July 2004

EXECUTIVE SUMMARY

This Report is a tutorial on the problems of establishing any-to-any 2-way real-time communications over the Internet. Such communications include separate signalling and media streams and frequently encounter problems where users are on a private network such as a LAN that is connected to the Internet via a firewall and/or a Network Address Translator.

Most of the issues will also be relevant to real-time communications over telco run Next Generation Networks (NGNs) since their services are expected also to be delivered to users over the same LANs as are used for Internet access.

The Report covers four issues:

- User identification
- Address assignment
- Network Address Translators
- Firewalls.

The four different types of Network Address Translators are explained, together with the way in which they affect real time communications.

Solutions for traversing Network Address Translators and Firewalls are explained with diagrams.

Understanding the issues covered in this Report is important for governments and regulators because:

> The development of techniques to traverse firewalls and network translators will be a major factor in the development of voice communications over the Internet and an influence on the market demand for IPv6, since IPv6 is seen as a means of removing the need for Network Address Translators.

INDEX TABLE

1 INTRODUCTION

This Report is a tutorial on the problems of establishing any-to-any 2-way real-time communications over the Internet. Such communications include separate signalling and media streams and frequently encounter problems where users are on a private network such as a LAN that is connected to the Internet via a firewall and/or a Network Address Translator.

Most of the issues will also be relevant to real-time communications over telco run Next Generation Networks (NGNs) since their services are expected also to be delivered to users over the same LANs as are used for Internet access.

The Report covers four issues:

- User identification
- Address assignment
- Network Address Translators
- Firewalls.

To establish real-time communications for services such as voice between two parties, A and B, each party first needs to establish some signalling communication with the other party and then they need to agree:

- The IP address and port number for each end of the media stream
- The codec to be used for the communications.

There are various obstacles to achieving these objectives:

- The IP addresses of each party may be assigned dynamically and so not be know permanently to the other party
- A and/or B may be behind a firewall which may block communications to ranges of IP addresses and port numbers
- A and/or B may be behind a NAT, which changes the IP addresses and port numbers in the packets.

This paper tries to explain these obstacles in a methodical way and discusses how they may be surmounted.

The issues covered in the paper are important because they are the controlling factor in the growth of real-time communications over the Internet and hence are a major factor in the future development of telecommunications and a factor in the case for IPv6. Some ISPs and mobile operators deploy NATs between their networks and the public Internet.

Acknowledgement: Alan Duric, formerly of GlobalIPSound, has contributed a great deal of information and advice that has assisted the preparation of this report.

2 USER IDENTIFICATION

The first issue of establishing real-time communications is identifying the party that you want to communicate with. In telephony this is done by using the telephone number (E.164 number) and the telephone networks are organized to make the user's relationship to his telephone number as stable and long lasting as possible.

The identifier used for routing packets to end points on the Internet is the public IPv4 address and the user's relationship to this address is much less stable and long lasting than that for his telephone number: the IP address used may change each time the user accesses the Internet. Consequently users are normally identified by an Internet name (eg user@domain) or an E.164 number and this identifier is translated to the current value of the IP address for use by the network.

To perform this translation and enable incoming communications a user needs to be registered with an entity in the public Internet and to inform that entity of its current IP address. This entity may be an "Instant Messenger" server or a SIP server (SIP- Session Initiation Protocol- is the protocol commonly used for real time communications). The user will have a stable identity such as a SIP address and for incoming communications, and the server translates the stable identity to the current IP address. Where ENUM is used, ENUM translates the user's E.164 number to the user's SIP address.

The Domain Name System (DNS) facility translates the domain part of the SIP address to the IP address of the SIP server to enable the calling entity to communicate with the SIP server. This is illustrated in figure 1.

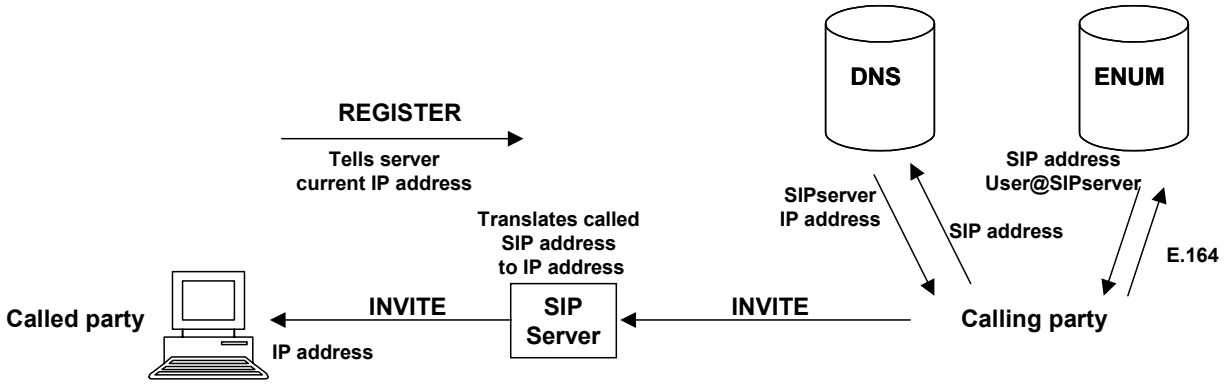

Figure 1: Translations for establishing incoming communications with SIP

The process for Instant Messaging is similar and is shown in figure 2

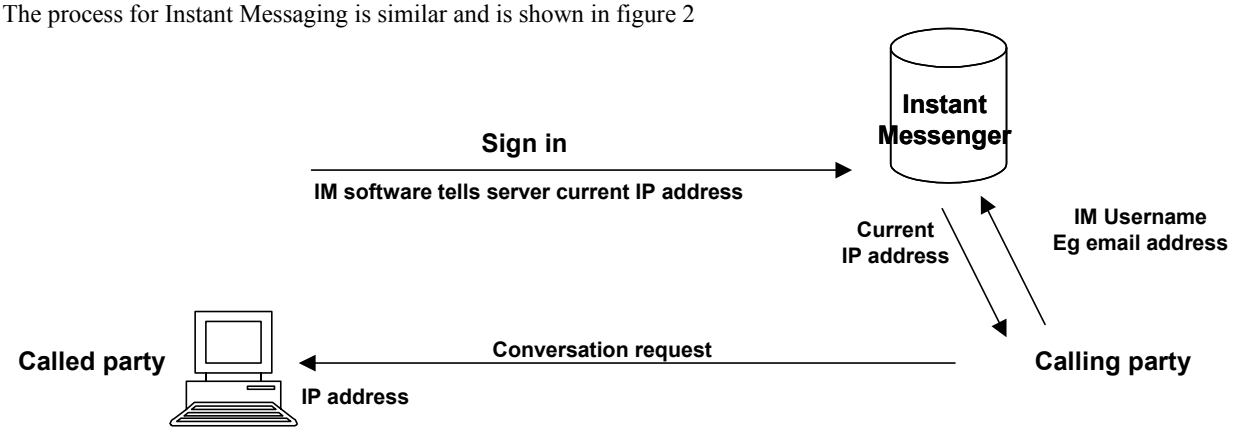

Figure 2: Translations for establishing incoming communications Instant Messaging

3 ADDRESS ASSIGNMENT

There are two types of IP address in IPv4:

- Public addresses, which are used in the public Internet and are globally unique. They are assigned under the control of ICANN via Regional Internet Registries and Internet Service Providers.

- Private addresses, which are local to the end system, eg the user's LAN. The following address rages are reserved for private use (see RFC 1918):
  - 10.0.0.0 - 10.255.255.255 (10/8 prefix)
  - 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
  - 192.168.0.0 - 192.168.255.255 (192.168/16 prefix).

A process in a computer that is to communicate over the Internet needs to be identified on the public Internet by a public IPv4 address and a port number. Port numbers are identifiers that are used to share IP addresses and they are organised in ranges that are allocated to different types of application (see RFC 1700 or www.iana.org).

IP addresses may be assigned to hosts (eg PCs) either permanently (by manual or automatic means) or dynamically. Dynamic assignment is used where the host may be connected or operational for only a small part of the overall time so that addresses can be re-used by other hosts when a host is no longer connected.

In LANs, desktop machines may be configured with permanent addresses whilst laptops have dynamic addresses because they are frequently connected and disconnected. Wireless access would also typically use dynamic allocation. Figure 3 shows some typical options.

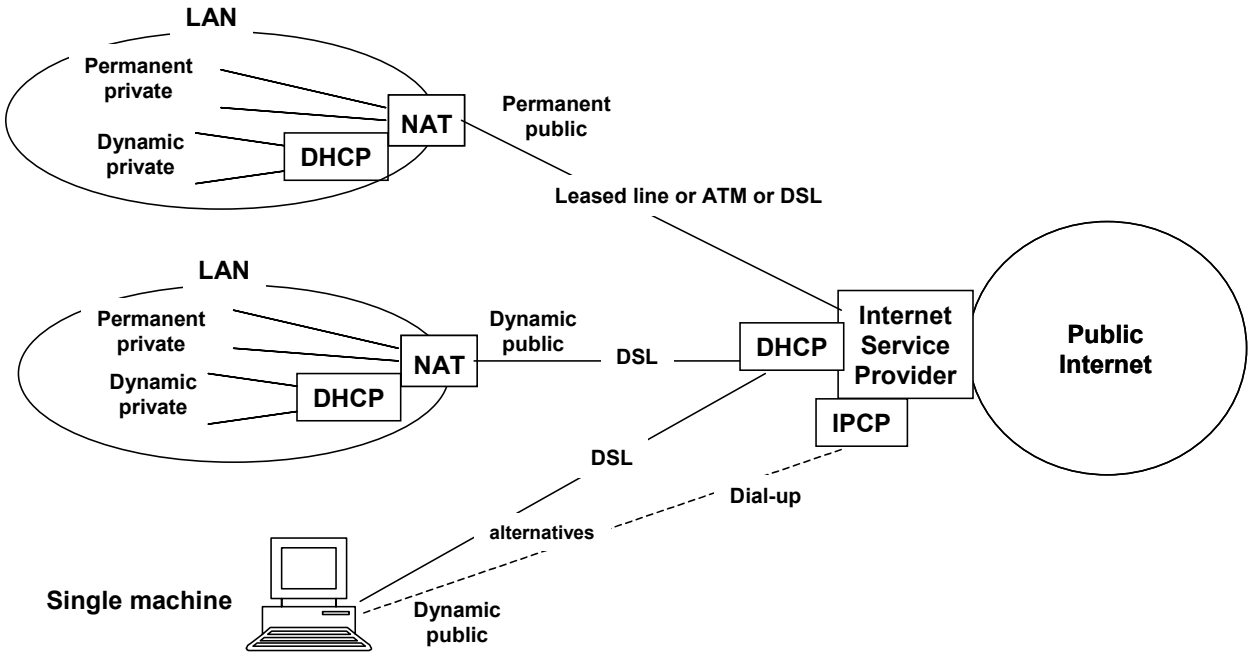

Figure 3: Typical options for address assignment

The Dynamic Host Configuration Protocol (DHCP) (RFC 2131) is the most common protocol for the assignment of IP addresses over permanent links such as leased lines and DSL.

DHCP works as follows:

1. The client host broadcasts a DHCPDISCOVER message on the LAN using the broadcast capabilities of the LAN such as are provided by Ethernet.
2. Each DHCP server responds with a DHCPOFFER message offering a particular free IP address. There may be more than one DHCP server on the LAN and more than one DHCPOFFER message may be received by the client host
3. The client host selects the IP address that it prefers (may be the first one offered) and sends a DHCPREQUEST message requesting that address from the server that offered it.
4. The server allocates the address in a DHCPACK message and the address is then used until the client host sends a DHCPRELEASE message to release it. If the address cannot be allocated because of some change in circumstances, the server sends a DHCPNAK message refusing the request and the whole process must be restarted.
5. Addresses are allocated for a fixed time and will be released if a DHCPRELEASE message has not been received by that time.

IP addresses on LANs will be private addresses if the LAN is behind a NAT.

For dial-up links, the dynamic address assignment is carried out by the Internet Protocol Control Protocol (IPCP) (RFC 1332) which sets up the network level configuration over the link established by the Point to Point Protocol (PPP) (RFC 1661). Dial-up arrangements can be quite complex with the client dialing a Network Access Server, authentication taking place between the Network Access Server and a remote RADIUS server and a Layer 2 tunnel being extended to a tunnel server at the entry into an Intranet. In these cases the dynamic address assignment is carried out by the tunnel server after authentication.

4 NETWORK ADDRESS TRANSLATION

4.1 Background

Network Address Translators (NATs) are devices that enable a small number of public IP addresses to be pooled and shared by a larger number of IP endpoints. They also protect the private network by hiding its internal addressing structure and topology. The IP endpoints inside the area served by a NAT have private IP addresses. Thus a NAT provides the translations between internal private addresses and external public addresses.

NATs work on the basis of communications sessions, which are identified uniquely by the combination of:

- The sending IP address and port number
- The destination IP address and port number.

When a IP endpoint (e.g. PC) in the private network sends packets to the Internet, the NAT device intercepts the packet and replaces the sending private IP address and port number by a public IP address and port number. Subsequently, it remembers this translation and when an incoming packet is received with the same public IP address and port number, it replaces them with the private IP address and port number and sends the packet into the private network. NATs enable several machines (IP endpoints) to share the same public (external) IP address simultaneously because NATs use port numbers to discriminate between sessions on the same machine.

NAT translation mappings (or "bindings") are typically initiated by outgoing communications and NATs block incoming communications for which there is no valid entry in the translation table. Certain servers are exceptions, eg email servers where a NAT usually supports a permanent relationship of a public address and well known port number to a server in the private network.

Because NATs change the values of IP addresses and port numbers in packets they interfere with the operation of applications (higher layer protocols) that are aware of IP addresses. Where the protocols are common, some NATs will alter the fields at the higher layer to maintain the match between these fields and the IP addresses and port numbers in the packets. Such functions are known as Application Layer Gateways (ALGs).

NATs are stateful, ie they remember information about sessions. Sessions are subject to timers that will cancel the session information if the session is inactive for certain period of time.

In practice the functions of NATs overlap those of firewalls in that NATs also provide some protection for machines on private networks.

4.2 Types of NAT

The following four types of NATs are used:

- Full Cone
- Restricted Cone
- Port Restricted Cone
- Symmetric.

The first three listed types of NAT maintain a mapping of a given internal address that is independent of the destination address being sought, so each internal IP:Port maps to a single external IP:Port. In contrast, the symmetric NAT allocates a new mapping for each separate destination address, and so each internal IP:Port may map to several different external IP:Ports. The mapping opens when the first packet is sent out from a IP endpoint through the NAT and remains valid for a limited time (typically a few minutes), unless packets continue to be sent and received on that IP:port. The exception is when the NAT has a static mapping table.

4.2.1 Full cone

For this NAT type, the mapping is well established and anyone from the public Internet that wants to reach a client behind a NAT, needs only to know the mapping scheme in order to send packets to it. For example, a computer behind a NAT with IP 192.168.0.5 sending and receiving on port 12345, is mapped to the external IP:port on the NAT of 195.242.54.253:5555. Anyone on the Internet can send packets to that IP:port and those packets will be passed on to the client machine listening on 192.168.0.5:5555.

4.2.2 Restricted Cone

In the case of a restricted cone NAT, the external IP:port pair is only opened up once the internal computer sends out data to a specific destination IP. For example, in the case where the client sends out a packet to external computer 1, the NAT maps the client's 10.0.1.5:24000 to 195.242.54.251:12345, and External 1 can send back packets to that destination. However, the NAT will block packets coming from External 2, until the client sends out a packet to External 2's IP address. Once that is done, both External 1 and External 2 can send packets back to the client, and they will both have the same mapping through the NAT.

4.2.3 Port restricted cone

The port restricted cone NAT will block all packets unless the client had previously sent out a packet to the IP and port pair that is sending to the NAT (otherwise, it is identical to a restricted cone). So, if the client has sent out packets to multiple IP:port pairs, they can all respond to the client, and all of them will respond to the same mapped IP:port on the NAT.

4.2.4 Symmetric

The symmetric NAT differs from the first three in that a specific mapping of internal IP:port to the NAT's public IP:port is dependent on the destination IP address that the packet is sent to. For instance, if the client sends from 10.0.1.5:12345 to Computer X, it may be mapped as 195.242.54.253:5555, whereas if the client sends from the same port (10.0.1.5:12345) to a Computer Y, it is mapped differently (195.242.54.253:5000).

Computer X can only respond to it's mapping (accordingly Computer Y can only respond to it's mapping). If either one tries to send to the other's mapped IP:port, those packets will be dropped.

Likewise as in the case of the restricted NAT, the external IP:port pair is only opened up once the internal IP endpoint sends out data to a specific destination.

The following table summarises the different types of NATs.

Type	Who can send and acceptable incoming packet?	NAT checks	Number of mappings held	State information and complexity
Full cone	Any public IP endpoint that knows the public IP:port	To: Public IP:port	One per sending private IP:port	Low
Restricted Cone	Any public IP endpoint that knows the public IP:port and has been sent an outgoing IP packet	To: Public IP:port From IP (checked separately)	One per sending private IP:port	Medium
Port restricted cone	Any public IP application that knows the public IP:port and has been sent an outgoing IP packet to the sending port	To: Public IP:port From IP:port (checked separately)	One per sending private IP:port	High
Symmetric	Any public IP application that knows the public IP:port and has been sent an outgoing IP packet to the sending port	To: Public IP:port From IP:port (correct combination only)	One for each external IP:port addressed by each sending private IP:port	Very high

Figure 4 illustrates the different types of cone and the protection that they give.

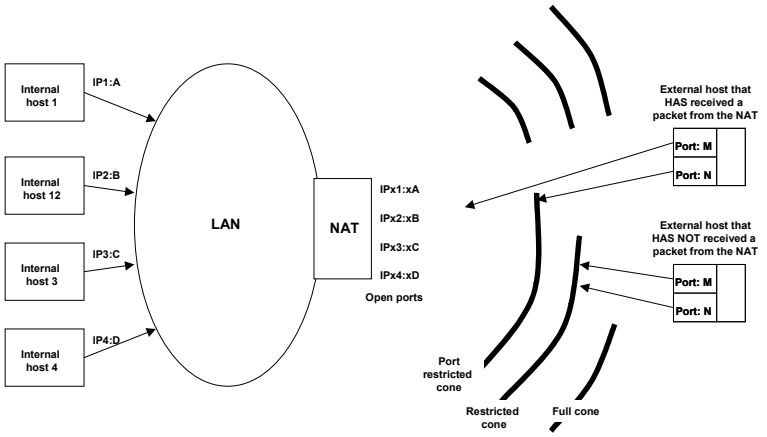

Figure 4: Different levels of protection from different types of cone NAT

With cone NATs, the external host can send to any internal host provided it knows the open IP:Port combination. For example, if internal host 1 has sent an out-going packet to external host J and internal host 2 has sent an out-going packet to external host K, then both J and K can send a packet to internal host 1 (and also to internal host 2) and provided that they pass the checks on the originating IP:Port the packets will pass through the NAT. With the symmetrical NAT, J can send only to internal host 1 and K can send only to internal host 2.

4.3 NAT and VoIP protocols

There are two separate processes in real-time communications:

- Registration (informing a server of the user's current IP address so that an external entity knows where the customer can receive incoming calls)
- Call establishment

4.3.1 Registration

Figure 5 shows registration where there is no NAT.

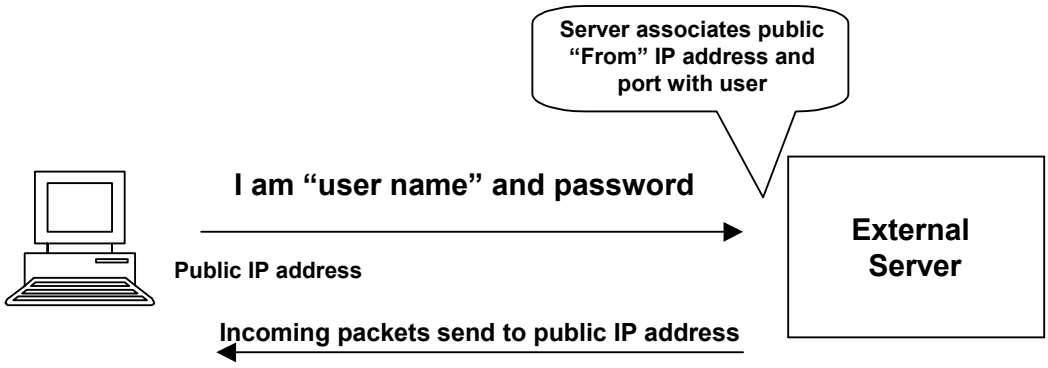

Figure 5: Registration without a NAT

The server sees the public "From" IP address and port number and associates it with the user for sending future incoming call set-up packets.

Figure 6 shows registration with a NAT. This normally works satisfactorily for all types of NAT provided the incoming packets are sent from the same IP address and port on the server that were used for the registration packet sent to the server. If a different IP address and port is used then the incoming packets will be rejected by the restricted cone, port restricted cone and symmetric types of NAT.

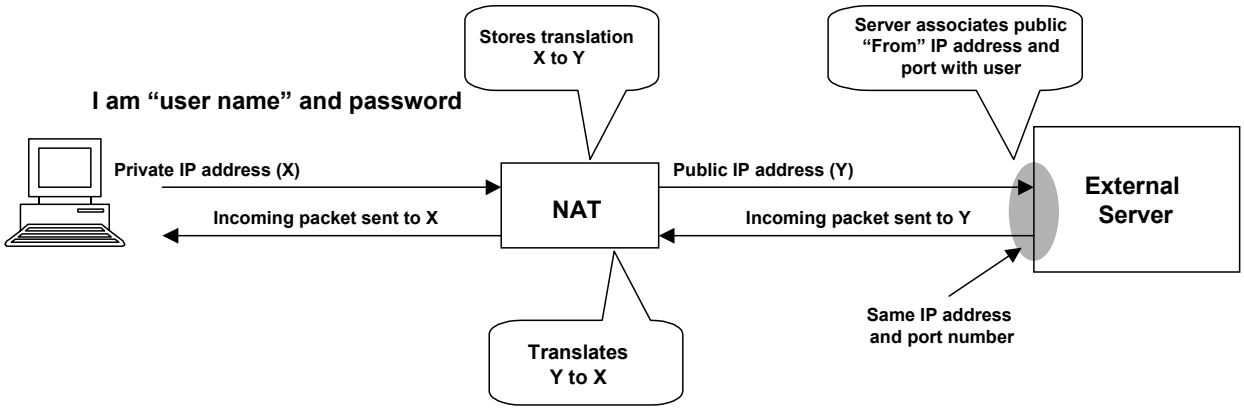

Figure 6: Registration with a NAT

4.3.2 Call establishment

Call establishment has two components:

- Signalling for call set-up
- Media for the voice packets.

The typical method of operation is that signalling takes place first between the calling and called parties and this signalling exchanges information on the IP address and port number to be used for the media channel. SIP is the most common protocol used for voice communications. Figure 7 shows the process in simplified form where there are no NATs.

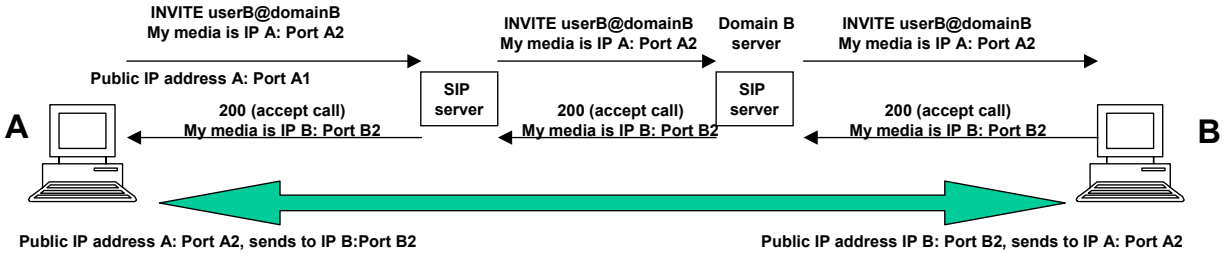

Figure 7: Simplified diagram of call establishment

Consider now the effect of a NAT at the A end. The situation is shown in figure 8. In this figure, the prefix "p" denotes "private" and the prefix "x" denotes "external".

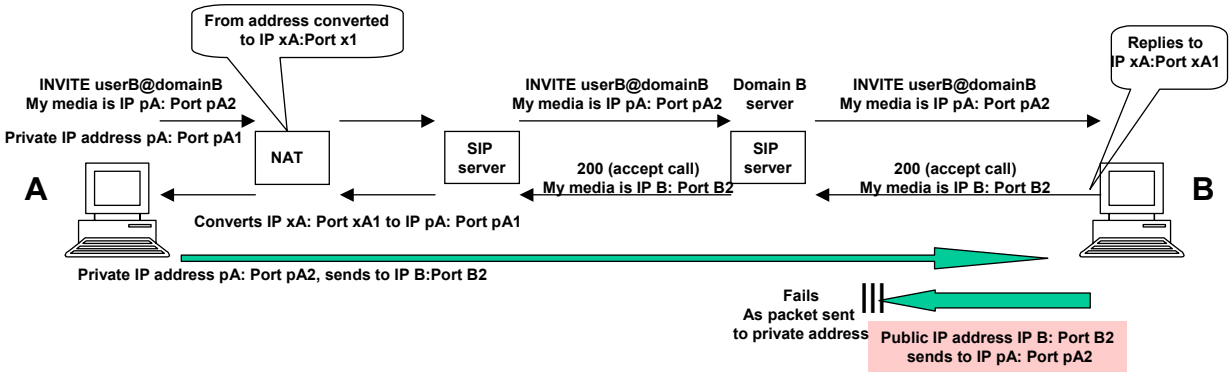

Figure 8: Simplified diagram of call establishment with the media path failing because of a NAT

The signalling works correctly in most cases. The SIP server, however, needs to return SIP packets to A from the same port on which it received them and send these packet to the IP address and port number on A that the packets were sent from (not to any standard SIP port, e.g. 5060). SIP has tags that tell the proxy to do this, so the "received" tag tells the proxy to return a packet to a specific IP address and the "rport" tag which port to return to return the packet to a specific port. Currently, a number of proxy implementations have not yet implemented the "rport" tag, and some clients will not parse the SIP messages correctly if these tags are present, but at least this mechanism is beginning to be used for NAT traversal.

The media, however, fails because the signalling from A tells B to send media packets to the private address of the media channel at A, and these private addresses are not supported on the Internet.

5 FIREWALLS

5.1 Types of firewall

A firewall is a network security device that ensures that all communications attempting to cross it satisfy the organization's security policy. Firewalls track and control communications, deciding whether to allow, reject or encrypt communications.

Most firewalls provide effective access control but many are not designed to detect and thwart attacks at the application level. In order to address the increasing threat from application-driven attacks, firewalls need to provide comprehensive security on multiple-levels to protect against both network and application attacks.

Historically, three different technologies have been used to implement firewalls:

- Packet Filters, usually implemented on routers, filter traffic based on packet content, such as IP addresses and port numbers. They examine a packet at the network layer and are application independent, which allows them to deliver good performance and scalability, but they are the least secure type of firewall. For example many firewalls are configured only to pass TCP or HTTP packets.

- Application-Layer Gateways, gateways use agents, called application proxies, to facilitate security by bringing context information into the decision process. Every application requires a new proxy, making scalability and support for new applications an issue. Therefore, Application-Layer Gateways tend to focus on providing either single application (e.g., web server) attack protection, or application access control without dedicated attack protection. For example, where the packet filter will only pass HTTP packets, an HTTP proxy might be used to provide a high degree of protection.

- Stateful Inspection has become common for high security. Stateful Inspection extracts the state-related information required for security decisions and maintains this information in dynamic state tables for evaluating subsequent connection attempts. This provides high level of security with very good scalability.

5.2 Firewalls and VoIP

Firewalls can be divided in two major groups: residential firewalls and corporate firewalls.

Residential and personal firewalls are easily configured to allow media to be transported through certain port numbers, but this presents a security risk. Residential firewalls do not normally implement application layer gateways or stateful inspection.

Corporate firewalls normally provide high security and open very few ports types (such as http, https, smtp, and pop3) and those ports are usually “subject to stateful inspection” or similar mechanisms. It is very unlikely that new ports (or range of ports) would be opened for VoIP connections (particularly inbound ones) on corporate firewalls, therefore solutions with media relays (see later) are the most likely ones to be successful. Paradial, by using HTTP tunnelling would pass fine stateful inspection and would not need any additional port to be opened. In contrast RTP relays would need certain ports to be opened and so require additional stringent security monitoring of those ports in order to ensure that only RTP media packets are allowed through.

6 TECHNIQUES FOR NAT AND FIREWALL TRAVERSAL

This section outlines some of the techniques that have been developed to enable real time communications with separate signalling and media streams to traverse NATs and firewalls.

If the client is behind a NAT that is not a symmetric NAT, then the solution for establishing the media channel correctly is fairly simple. The client A must find out what its internal media port looks like from the public Internet (i.e. it must determine the NAT mapping) and then it must put the external public IP address:port number into the signalling message instead of the internal private IP address:port number. There are two methods for a client to determine the NAT mapped public IP address:port number. The first is to ask the NAT, the second is to ask an entity outside the NAT on the public Internet. We now look in more detail at some of these techniques.

6.1 Universal Plug and Play (UPnP)

A client can ask the NAT how it would map a particular IP address:port number through a protocol called Universal Plug and Play (UPnP). This is a solution that is being embraced by UPnP consortium (www.upnp.org), and is being widely deployed by Microsoft and others. The client queries the NAT via the UPnP protocol asking what mapping it should use if it wants to receive on port x. The NAT responds with the IP address:port number pair that someone on the public Internet should use to reach the client on that port. Many NAT device manufacturers have already included UPnP in their products.

One problem with UPnP is that it will not work in the case of cascaded NATs (e.g. when ISP uses and NAT and provides customers with a private IP address, and customers also use a NAT to connect all devices to the ISPs network). There are also security issues that have not yet been addressed with UPnP, therefore UPnP is not being deployed much within organisations and is seen as a credible only for the residential market. Additionally, there is a problem because there is a large installed base of existing NATs that do not support UPnP.

6.2 External entities

In the absence of a method of communicating with the NAT device, the next best way for a client to determine its external IP address:port number is to ask a server (called a NAT probe) sitting outside the NAT on the public Internet what it sees as the source of a packet coming from this client. When the NAT probe receives a packet, it returns a message from the same port to the source of the received packet containing the IP address:port number that it sees as the source of that packet. In every case (all 4 NAT cases), the client will receive the return packet. The client can then determine:

- If it is behind a NAT (if the IP address:port number contained within the return packet is different from the IP address:port number used to send the packet).
- Which public IP address:port number it should send in the signalling messages in order for the B party to send media packets back.

Figure 9 shows how this works.

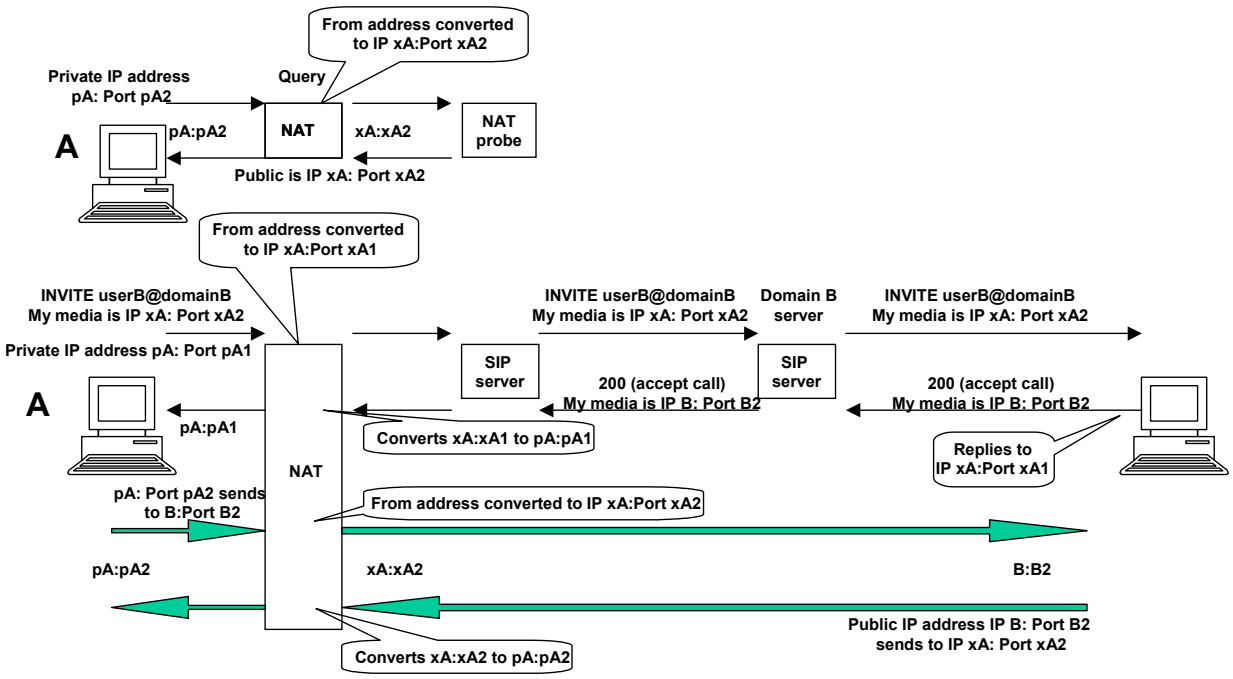

Figure 9: Simplified diagram of call establishment with the media path working through use of a NAT probe

For example, if the client wants to be reached on private address IP pA: Port pA2, it will first send out a query to the NAT probe from IP pA: Port pA2. The NAT probe will respond to that IP:port with a packet containing the public address IP xA: Port xA2 that will correspond to the private address. The client then puts into the SIP signalling "m= AUDIO xA2" and "c=xA" indicating that media should be sent to public address IP xA: Port xA2 while the client itself listens on private IP pA: Port pA2.

This will work when the following conditions are net:

- The client must send and receive media on the same port.
- The client must query the NAT probe shortly before sending out the SIP message otherwise the NAT mapping may time out and be changed.
- In the case of Restricted Cone or Port Restricted Cone NATs, the client must send out a media packet to B before the NAT will allow packets from B through to the client.

This solution will not work in the case of symmetric NATs, since the IP address of the NAT probe is different from that of B, and therefore the mapping the NAT probe sees is different from the mapping that the NAT would give to B.

6.3 Simple Traversal of UDP Through NATs (STUN)

Simple Traversal of UDP Through NATs (STUN) is a protocol (defined by IETF RFC 3489) for setting up the kind of NAT Probe that has been described in the previous section. STUN not only discovers and returns the public IP:port, but it also enables the client to determine the type of NAT used. A number of client software packages are already being developed that use STUN to enable them to configure their SIP/SDP messages to work with NATs. STUN can work where there is more than one NAT in tandem, but does not work for communications with a peer that is behind the same NAT, nor does it traverse a symmetrical NAT.

The STUN client is typically embedded in an application which needs to obtain a public IP address and port that can be used to receive data. The STUN client sends a Binding Request over UDP to the STUN server. STUN servers can be discovered through DNS SRV records, and the client normally knows the domain to use to find the STUN server. When a Binding Request arrives at the STUN server, it may have passed through one or more NATs between the STUN client and the STUN server. As a result, the source address of the request received by the server will be the mapped address created by the NAT closest to the server. The STUN server copies that source IP address and port into a STUN Binding Response, and sends it back to the source IP address and port of the STUN request.

Using a combination of different requests to a STUN server, a client can determine:

- If it is on the open Internet;
- if it is behind a firewall that blocks UDP;
- If it is behind a NAT, and what type of NAT it is behind.

Annex A explains the automatic detection of NATs and firewalls.

6.4 Solutions for symmetrical NATs

The NAT probe or STUN server will work only for the first three types of NAT. Symmetric NATs cannot be traversed by this scheme because they have apply mappings (bindings) depending on the target IP address. Consequently the mapping that the NAT applies between the client and the NAT probe will be different from the mapping that applies between the client and its peer.

In the case of a symmetric NAT, the client must send media packets to, and receive media packets from the same IP address. Even if a SIP connection has already been established, the endpoint outside the NAT must wait until it receives a packet from the client inside the NAT before it can know where to reply. This is known as Connection Oriented Media.

If an external endpoint is meant to communicate both with clients that are behind NATs and with clients on the open Internet, then it must know when it can use the IP address:Port identities for sending back media packets that it receives in the SIP message, and when it needs to wait until it receives a media packet directly from the client before it starts sending media packets back to the source IP:port of the media packets it receives.

One proposed solution in an IETF draft (draft-ietf-mmusic-sdp-comedia-05.txt) adds a line to the SDP in the SIP message (coming from the client behind the NAT) to inform the endpoint to wait for the incoming media packet. When the endpoint reads this line, it understands that the initiating client will "actively" set up the IP:port to which the endpoint should return media packets, and that the IP:port found in the SDP should be ignored.

If an endpoint supports Connection Oriented Media, then it can traverse symmetric NATs. However, two scenarios are still problematic:

- If the endpoint does not support Connection Oriented Media
- If both endpoints are behind Symmetric NATs.

In these cases the solution is to use a media relay (RTP relay) in the middle of the media packet flow between endpoints. Typically, there would be a server in the middle of the SIP flow, usually referred a NAT Proxy, that would manipulate the SDP in such a way as to instruct the endpoints to send media packets to the RTP Relay instead of directly to each other. The Relay would set up its own internal mapping of a session, noting the source IP:port of each endpoint sending it RTP packets. It then forwards the packets that is receives replacing the source IP:Port with the IP:Port identities recognised by the destination endpoint, and it does this for both directions of flow.

This solution will work for all types of NATs, but because of the delay associated with the RTP Relay (which may be substantial if the RTP Relay is not close to at least one of the endpoints), it should probably be used only when a Symmetric NAT is involved. In other NAT scenarios, modification of the IP:Port identity will be sufficient.

Because the client will not hear any voice until the first packet is sent to the RTP Relay, there could be problems for network announcements that need to be sent before the media streams are established.

6.5 Solutions for Firewalls and NATs

Firewalls typically block media packet types such as UDP and so the traversal solution is to use TCP tunnelling and relays for media in order to provide NAT and firewall traversal. Current solutions include:

- Tunnelling the media packets within TCP or HTTP packets to a relay. This solution is used by Paradial (www.paradial.com) and is called "Real-Tunnel". This solution uses additional functionality that operates in conjunction with SIP (or other real time communications clients) and packages the media packets into a TCP stream and sends the TCP packets to the relay. The relay then extracts the packets and send them on to the other endpoint. If the other endpoint is behind a symmetrical NAT or corporate firewall that does not allow VOIP traffic, the relay would transfer the packets to another tunnel. TCP was not designed for real time traffic such as voice, so a number of optimizations have been made by Paradial to adapt it to the needs of real time communications. This solution is effective but introduces additional delay and requires more bandwidth because of the tunnelling overhead.

- Skype (www.skype.com) wraps media packets in TCP and delivers good quality. Skype uses temporary relays that run on the machines of other users and minimises delay by choosing a machine that is close to the endpoint. It also uses advanced playout controllers to reduce effects of the additionally introduced delay. Skype uses a proprietary protocol and so does not interoperate with SIP endpoints. More information on Skype is given in Annex B.

ANNEX A: AUTOMATIC DETECTION OF NATS AND FIREWALLS USING STUN

Four tests are performed to determine the environment of the client. Assume that there are two STUN servers available, IP1 and IP2, and they can return responses either from port A or port B. The Binding Request messages carry "Change IP" and "Change Port" flags that determine whether the response should be sent from IP1 or IP2 or from Port A or Port B. Sending a request to IP1 without the Change IP or Change Port flags set will cause the STUN server to respond from IP1, port A. Setting the Change IP flag or Change Port flag will request a response from IP2 or Port 2 respectively. Figure 10 shows the tests in the order in which they are performed.

Test	Request sent to	Change IP	Change Port	Response from
Test 1	***IP1:A***	*N*	*N*	***IP1:A***
Test 2	***IP1:A***	*Y*	***Y***	***IP2:B***
Test 3	***IP2:A***	*N*	*N*	***IP2:A***
Test 4	***IP1:A***	*N*	*Y*	***IP1:B***

Figure 10: NAT tests

Figure 11 shows the test configurations.

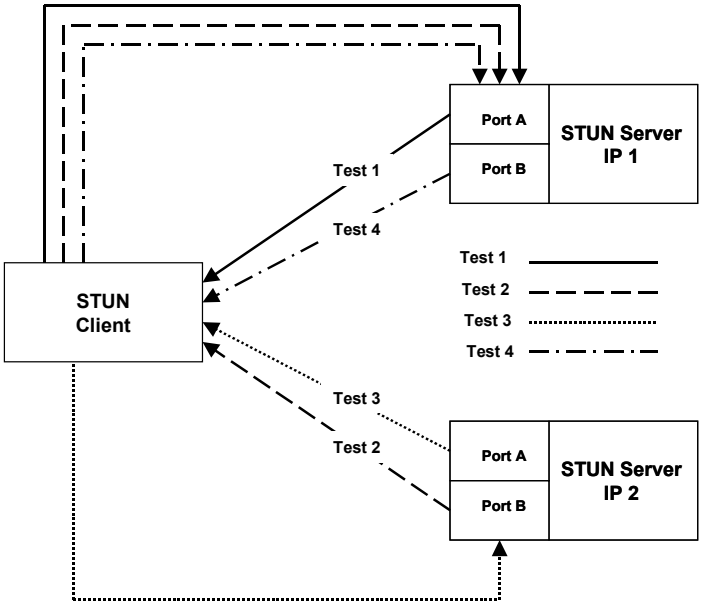

Figure 11: NAT test configurations

Figure 12 shows the logic of the test campaign and the information derived.

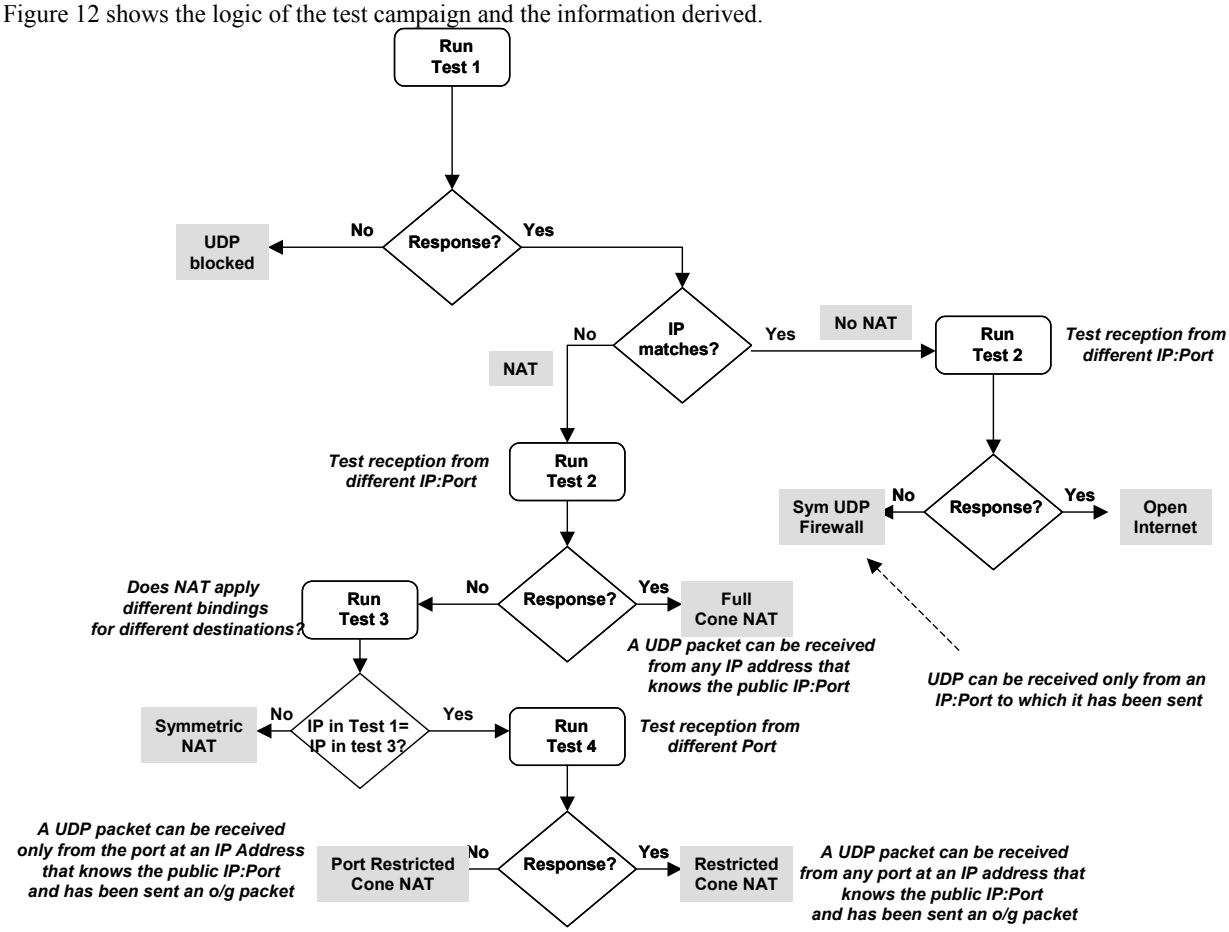

Figure 12: NAT test logic

When Test 1 is performed, if no response is received, then the client knows it is behind a firewall that blocks UDP and traversal is not possible.

If a response is received, the IP address in the Binding Response is tested against what the client thinks its IP address is. If the IP addresses in Test 1 match, Test 2 (Change IP and Port) is run. The response in then sent from a different IP:Port.

If the response is not received, then the client is behind a symmetric UDP firewall - that is its IP address is not behind a NAT, but its firewall will only allow UDP in from a given destination once the client has sent a packet out to that destination.
If the client receives the response, then the client knows that it is on the open Internet without a firewall blocking UDP.

If the IP addresses in Test 1 are not the same, Test 2 is also run with the response sent from a different IP:Port. If the client receives the response, then it is behind a Full Cone NAT and can receive a UDP packet from any source that knows the correct IP:Port. If no response is received, the client runs Test 3 and compares the IP address that is returned in the Binding Response against the IP address that was returned in Test 1 (from IP 1).

If the two IP addresses are not the same, then the client is behind a Symmetric NAT, which is applying different bindings to different destinations.

If the two IP addresses are the same, the client runs Test 4 (Change Port). If the response is received, then the client is behind a Restricted NAT, and can receive a UDP packet from any port at any IP address source that knows the correct IP:Port and has received an outgoing packet.

If the response is not received, then the client is behind a Port Restricted NAT, and can receive a UDP packet only from the port at an IP address source that knows the correct IP:Port and has received an outgoing packet.

ANNEX B: SKYPE

To a user, Skype looks like a simple Instant Messenger service. It supports only text messaging and voice, but it offers better voice quality than most of the IMS services and also works across most residential firewalls. The intention is to add a conferencing capability and PSTN breakout in the near future.

From a network perspective, Skype is completely different from other Instant Messenger services. It is designed to be a distributed (peer-peer) network and has been developed by the people who produced KaZaA, which is peer-peer file sharing software that is similar in function to Napster.

The Skype network is based on the Gnutella network concept that has been the subject of a number of academic papers. The aim is to make the communications as peer-peer as possible, but this is not simple because the IP addresses of the users frequently change due to dynamic address allocation and to users logging on in different places. Consequently some external function is needed to link users together. In IM and SIP systems, this is achieved by using fixed central servers; users register their current location and IP address when the log on to the system.

With Skype, most of the central server functions are replaced by a varying number of supernodes that are running on users machines typically without the user's knowledge. The supernodes hold and exchange data about who is on-line and where they are currently located. The central server does not hold any user data but holds only the IP addresses of the supernodes. Users machines are assessed for suitability as supernodes according to their time on-line, the machine power, and the absence of a NAT and firewall. Users may be unaware that their machines are working as supernodes. Typically there is about one supernode for 500 users. Thus the management system is expanded and contracted according to the degree of use, and the management functions are located relatively close to the users that they are managing. The design is thus made scaleable.

When a user logs on, the Skype software on their machine attempts to locate and register with a supernode. The software uses a cache of supernode IP addresses collected from its previous sessions but can contact a central server if a local supernode cannot be found.

When a user attempts to make a call, the software contacts the supernode and the supernode finds the called user by contacting the other supernodes as necessary. The end users then negotiate directly with each other to establish the media channel.

As soon as the end users have located each other, all subsequent communications are as direct as possible. If neither end user is behind a firewall, these communications will be true peer-peer. Where a user is behind a firewall so that all incoming communications have to be preceded by an outgoing communication, the firewall may inhibit peer-peer communications and it may be necessary to route the communications through a relay. Other users' machines close to the end users are used as relays. Supernodes are not normally used as relays. Figure 13 shows the basic arrangement for Skype.

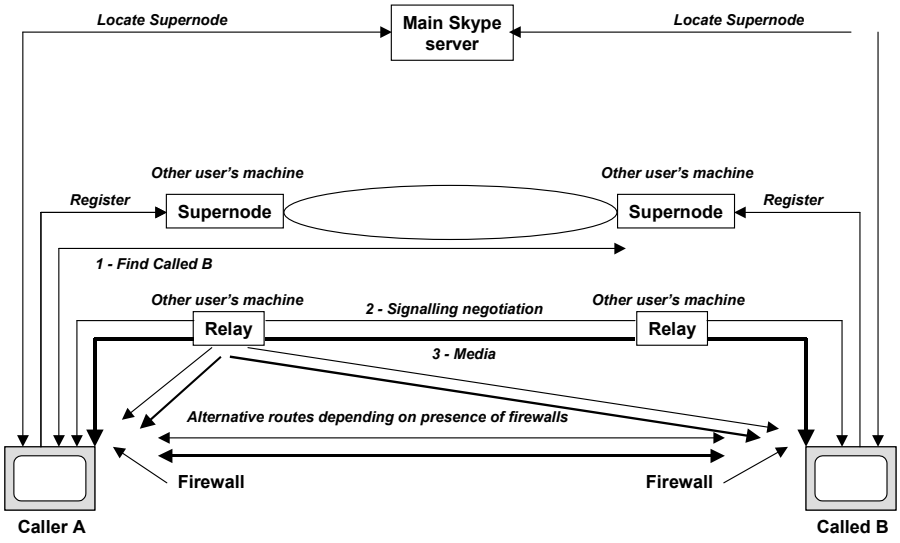

Figure 13: Skype

Where a relay is used, the relay may go off-line during the call. Skype establishes several alternative routes to use as hot spares if this happens.

Two features give Skype an advantage over most other Instant Messenger services:

- Skype uses new advanced codecs developed by GlobalIPSound that are tolerant to packet loss and give wideband voice quality. Skype software includes four of these codecs as well as traditional telco codecs and selects the most suitable codec based on an assessment of the media path and the end user machine's capabilities.

- Skype uses intelligent methods to traverse firewalls and NATs and applies the method that is most appropriate to the end user's situation. The methods include routing via relay, encapsulation in a TCP tunnel and STUN type protocols. The net result is that Skype can work in most circumstances that are typical of residential and small business use, but it cannot yet handle the more stringent security provided by some larger companies.

Skype includes both authentication and strong encryption of both the text messages and the media streams. The encryption uses public key techniques where the key pairs are generated by the software in the end systems and are not known to the server, supernodes or relays. The encryption algorithm is the Advanced Encryption Standard (AES), which is used by the US Government. Skype needs encryption because the media may pass through the machines of other users and because Skype will be used in future from pocket devices over WiFi access.

ANNEX C: ABBREVIATIONS

AES	Advanced Encryption Standard
ALG	Application Layer Gateway
DHCP	Dynamic Host Configuration Protocol (RFC 2131)
DNS	Domain Name System (RFC 1035)
ENUM	E.164 number and DNS (RFC 2916)
HTTP	Hypertext Transfer Protocol (RFC 2616)
ICANN	Internet Corporation for Assigned Names and Numbers
IETF	Internet Engineering Task Force
IMS	Instant Messenger Service
IP	Internet Protocol (RFC 0791)
IPCP	Internet Protocol Control Protocol (RFC 1332)
LAN	Local Area Network
NAT	Networks Address Translator
NGN	Next Generation Network
PC	Personal Computer
PPP	Point to Point Protocol (RFC 1661)
PSTN	Public Switched telephone Network
RADIUS	Remote Authentication Dial In User Service (RFC 2865)
RFC	Request For Comment
RTP	Real Time Protocol (RFC 3350)
SDP	Session Description Protocol (RFC 2327)
SIP	Session Initiation Protocol (RFC 3265)
SMTP	Simple Mail Transfer Protocol (RFC 2821)
STUN	Simple Traversal of User Datagram Protocol Through Network Address Translators (RFC 3489)
TCP	Transmission Control Protocol (RFC 793)
UDP	User Datagram Protocol (RFC 768)
UPnP	Universal Plug and Play
WiFi	Wireless Fidelity