

SIEMENS

MySip.ch

SIP
Network Address Translation (NAT)

SIP Architecture with NAT
Version 1.0

Issued by
DS MS, Software house
Albisriederstr. 245, CH-8047 Zurich

Copyright © Siemens Schweiz AG 2004

All Rights Reserved.

SIEMENS SCHWEIZ AKTIENGESELLSCHAFT

Author: P.Mächler
Translator:

DS MS

In addition to the authors named on the cover page, the following persons have collaborated on this document:

Shaun Baker

DS MS

The document comprises 27 pages, all pages have issue no 01.

The document is based on template RSPEC.DOT.

This issue was last saved on 06.05.2004 10:44.

This document was edited with MS WinWord version 8.0b.

The filename of the main document file was

J:\PROJECTS\IDS_MS\SIP\General_Documents\NAT\SIP_Architecture_with_NAT_V1.0.doc.

0 General Information	4
0.1 Issue Control	4
0.2 History	4
0.3 References	4
0.4 Glossary and Abbreviations	4
0.5 Keyword/Descriptor	5
0.6 List of Figures and Tables	5
1 Introduction	6
1.1 About this document.....	6
1.2 Network Address Translation (NAT).....	6
1.2.1 What is NAT	6
1.2.2 NAT types	7
1.2.2.1 Full Cone.....	7
1.2.2.2 Restricted Cone.....	8
1.2.2.3 Port Restricted Cone	8
1.2.2.4 Symmetric.....	9
2 Techniques for NAT	10
2.1 UPnP.....	10
2.2 Application Layer Gateway (ALG).....	11
2.3 External Query.....	12
2.4 STUN.....	13
2.4.1 Automatic Detection of NAT Environment	13
2.5 Connection Oriented Media (Comedia)	16
2.6 RTP Relay (TURN)	17
2.6.1 Call Flow.....	18
2.7 Inbound call	19
3 Analysis of different Solutions	20
3.1 UPnP.....	20
3.2 STUN (External Query).....	21
3.3 Application Layer Gateways (ALGs).....	22
3.4 Connection Oriented Media (Comedia)	23
3.5 RTP Relay (TURN)	24
4 Recommended architecture	25
4.1 Architecture, 1 st step: Comedia and TURN	25
4.2 Architecture, 2 nd step: ALGs	26
4.3 Architecture, 3 rd step: UPnP	26
4.4 Architecture, 4 rd step: STUN.....	27

0 General Information

0.1 Issue Control

The document comprises 27 pages, all pages have issue no 02.

0.2 History

Issue	Date	Reason for Changes
02	27 Feb. 04	Corrections by Shaun Baker
01	25 Feb. 04	Creation

Table 1: History

0.3 References

- /1/ NAT Traversal in SIP
DeltaThree
www.deltathree.com
- /2/ Short Term NAT Requirements for UDP Based Peer-to-Peer Applications
IETF Draft, C.
Hitema, Feb. 2001.
- /3/ "STUN – Simple Traversal of UDP Through NATs"
ETF Draft, draft-rosenberg-midcom-stun-00
Rosenberg, Weinberger, Huitema, Mahy
October 1, 2001.
- /4/ "Connection-Oriented Media Transport in SDP"
IETF Draft, draft-ietf-mmusic-sdp-comedia-05
D. Yon
March 2003.
- /5/ Traversal Using Relay NAT (TURN)
draft-rosenberg-midcom-turn-01
Rosenberg, Weinberger, Huitema, Mahy
March 2003.

0.4 Glossary and Abbreviations

NAT	N et A dress T ranslation
STUN	S imple T raversal of U DP through N ATs
UPnP	U niversal P lug'n P lay
TURN	T raversal U sing R elay N AT
ALG	A pplication L ayer G ateway

0.5 Keyword/Descriptor

0.6 List of Figures and Tables

Figure 1: NAT schematic.....	6
Figure 2: Full Cone NAT.....	7
Figure 3: Restricted Cone NAT.....	8
Figure 4: Port Restricted Cone NAT	8
Figure 5: UPnP.....	10
Figure 6: Application Layer Gateway (ALG).....	11
Figure 7: External Query	12
Figure 8: STUN	14
Figure 9: Connection Oriented Media (Comedia).....	16
Figure 10: RTP Relay	17
Figure 11: Involved Components for UPnP.....	20
Figure 12: Involved Components for STUN	21
Figure 13: Involved Components for ALGs.....	22
Figure 14: Involved Components for Comedia.....	23
Figure 15: Involved Components for the RTP Relay(TURN).....	24
Figure 16: Architecture, 1 st step: Comedia and RTP Relay.....	25
Figure 17: Architecture, 2 nd step: ALGs	26
Figure 18: Architecture, 3 rd step: UPnP	26
Figure 19: Architecture, 4 rd step: STUN.....	27
Table 1: History	4
Table 2: STUN Tests.....	13
Table 3: Analysis of UPnP	20
Table 4: Analysis of STUN	21
Table 5: Analysis of ALGs	22
Table 6: Analysis of Comedia.....	23
Table 7: Analysis of the RTP Relay	24

1 Introduction

1.1 About this document

This document describes the general requirements for establishing a SIP infrastructure over the Internet. The main focus is on access technologies such as ADSL modems, ADSL Routers and Firewalls.

The aim of this document is to show an architecture capable of supporting most of these access technologies.

1.2 Network Address Translation (NAT)

NAT is a big problem for VoIP communication, mainly because IP addresses are exchanged within higher-level protocols (H.323 & SIP). There are several strategies solving this problem, but none of them is a complete or satisfying solution. At the moment, a combination of several strategies/implementations provides be the best solution.

1.2.1 What is NAT

Many service providers and private individuals are using network Address Translation (NAT) as a way to get around the problem of not having enough IP addresses. An enterprise may have a block of IP addresses assigned to them, but many more computers than the allocated IP addresses. Also an individual may have a DSL connection with one IP address, but want to have multiple computers hooked up to the Internet. NAT solves this problem by mapping internal addresses to external or public addresses. An internal IP address:port pair is mapped to an external IP:port, and whenever the NAT receives a packet with the external IP:port, it knows how to reroute the packet back to the internal IP address and port. The mapping is valid for some predefined mapping interval after which, in the absence of network traffic between the two communicating parties, this mapping may be purged. In all cases, we assume that an application will send and receive packets on the same port. Many people also use NAT as their first line of defense against malicious intrusion attempts.

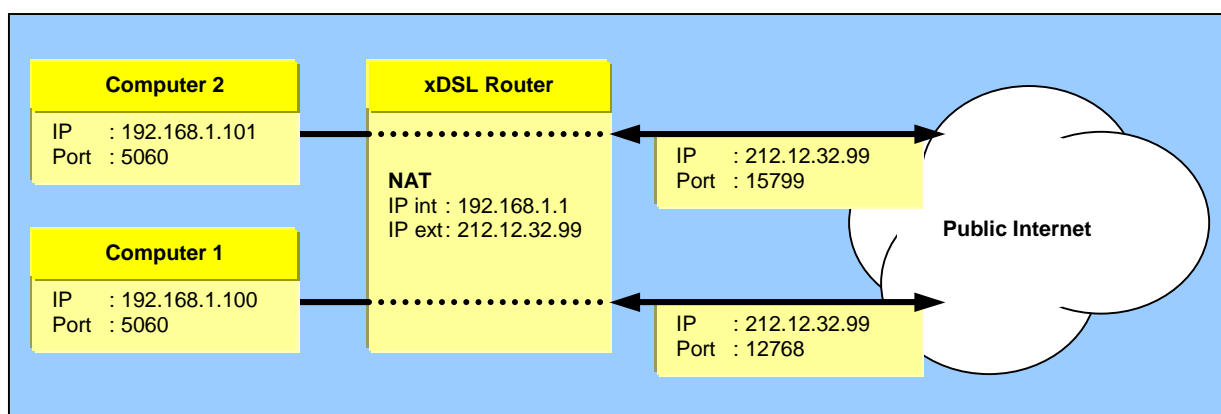


Figure 1: NAT schematic

1.2.2 NAT types

There are four types of NATs. As defined in /2/ they are:

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

For a given internal address, the first three types of NAT maintain a mapping of this internal address that is independent of the destination address being sought. The fourth type of NAT will allocate new mappings that are dependent on destination address.

Unless the NAT has a static mapping table, the mapping that opens when the first packet is sent out from a client through the NAT may only be valid for a certain amount of time (typically a few minutes), unless packets continue to be sent and received on that IP:port.

1.2.2.1 Full Cone

In the case of the full cone, 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 192.168.1.100 sending and receiving on port 5060, is mapped to the external IP:port on the NAT of 212.12.32.99:12768. Anyone on the Internet can send packets to that IP:port and those packets will be passed on to the client machine listening on 192.168.1.100:5060.

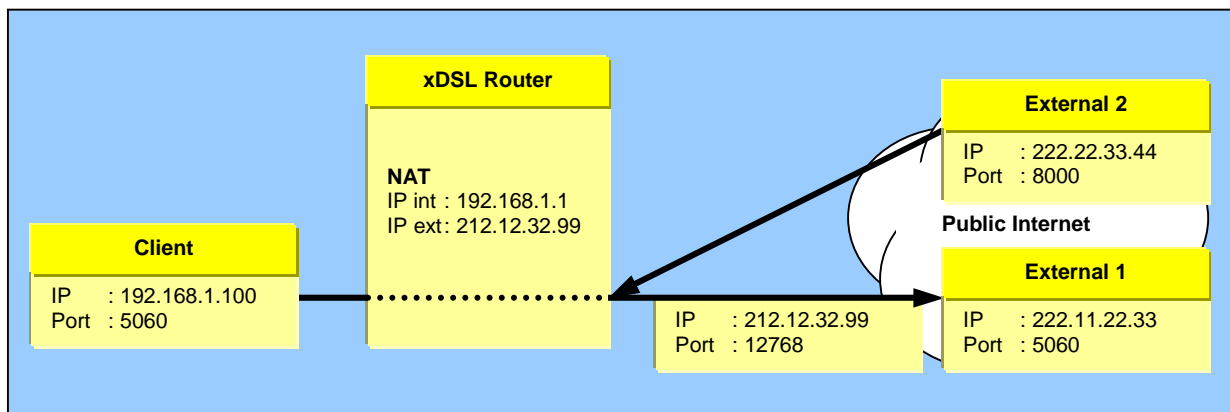


Figure 2: Full Cone NAT

1.2.2.2 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 10.0.0.1:8000 to 202.123.211.25:12345, 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.

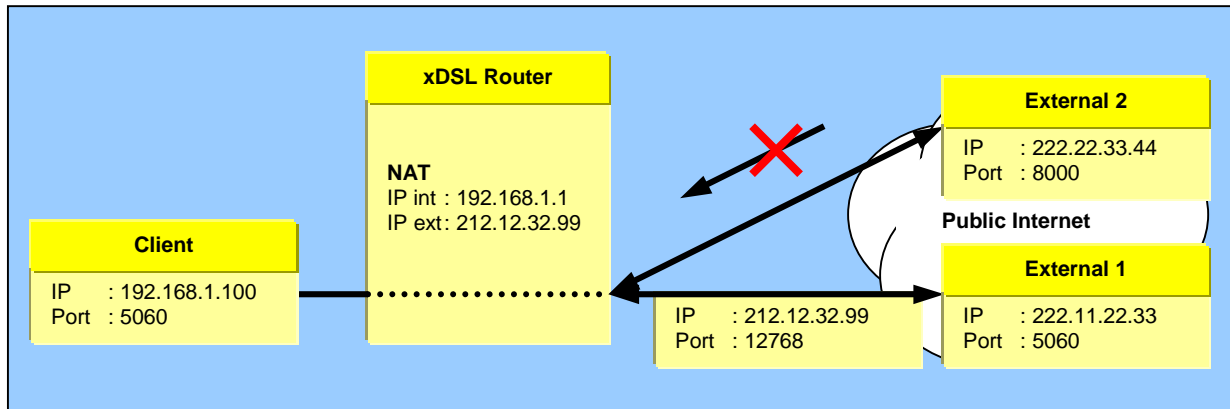


Figure 3: Restricted Cone NAT

1.2.2.3 Port Restricted Cone

A port restricted cone type NAT is almost identical to a restricted cone, but in this case the NAT will block all packets unless the client had previously sent out a packet to the IP AND port that is sending to the NAT. So if the client sends to External 1 to port 5060, the NAT will only allow through packets to the client that come from 222.11.22.33:5060. Again, 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.

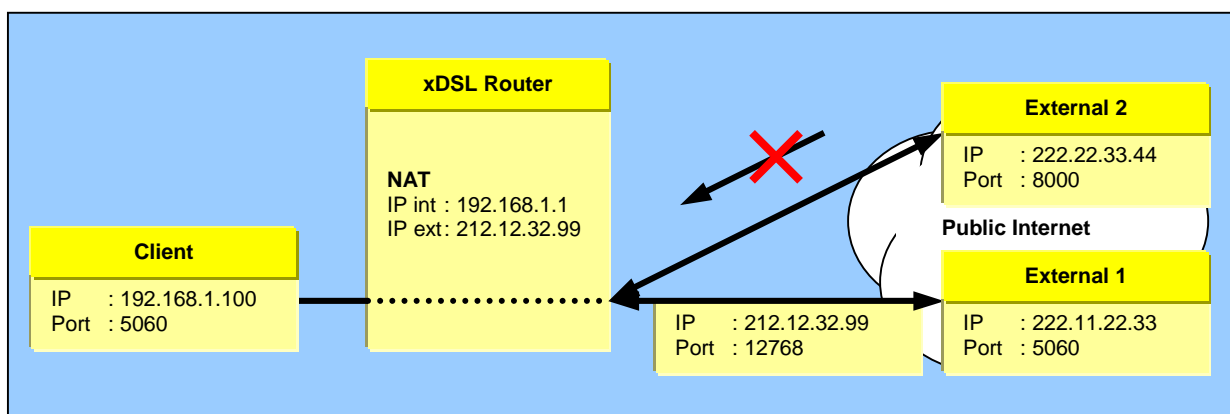
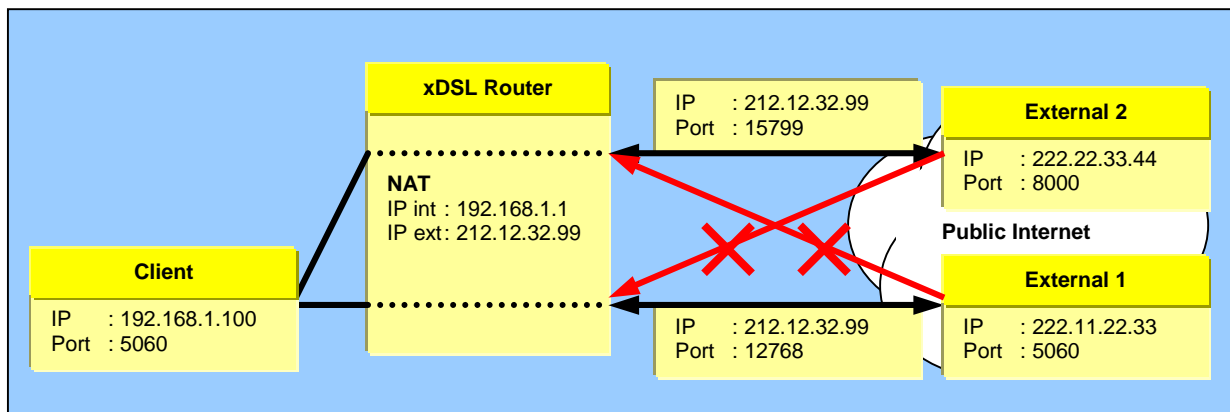


Figure 4: Port Restricted Cone NAT

1.2.2.4 Symmetric

The last type of NAT – symmetric - is different from the first three in that a specific mapping of internal IP:port to the NAT's public IP:port is dependant on the destination IP address that the packet is sent to. So for example, if the client sends from 192.168.1.100:5060 to External 2, it may be mapped as 212.12.32.99:15799, whereas if the client sends from the same port (192.168.1.100:5060) to a different IP, it is mapped differently (212.12.32.99:12768). External 2 can only respond to its mapping and External 1 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. As in the case of the restricted NAT, the external IP:port pair is only opened up once the internal computer sends out data to a specific destination.



2 Techniques for NAT

2.1 UPnP

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

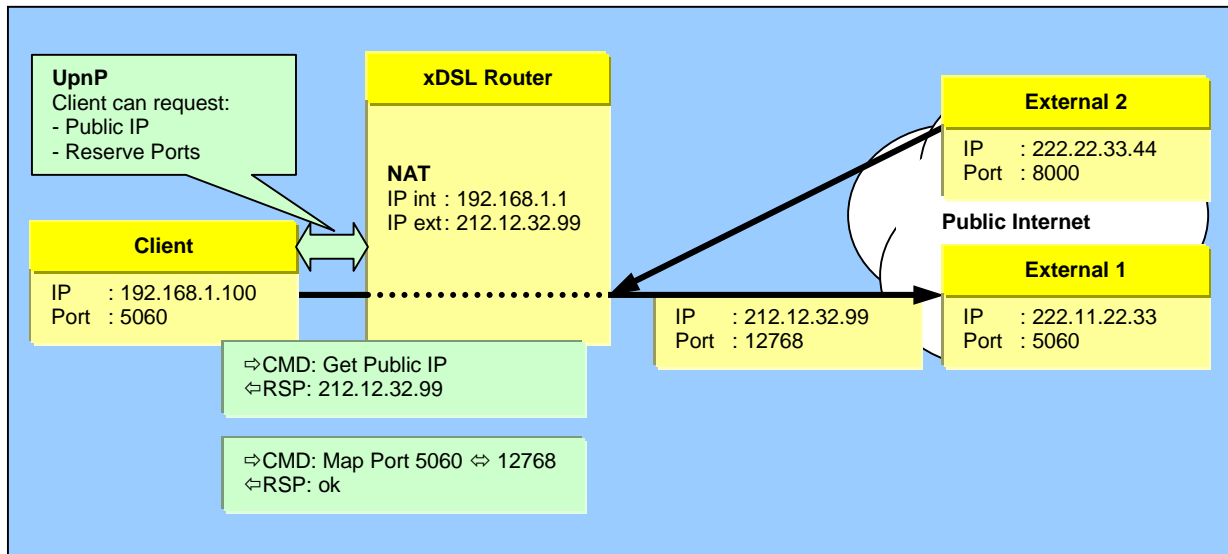


Figure 5: UPnP

With UPnP the client is able to steer the Router, and will therefore be able to modify the SIP/SDP messages in the correct way.

One problem with UPnP is that it will not work in the case of cascading NATs. For example, say an ISP owns a block of IP addresses, but not enough to service its user base. The ISP would use a NAT to provide IP addresses to its customers. One of those customers may require many IP addresses (for example, an internet café) so it would set up its own NAT to share its one address between many computers. If a client running on one of the local computers were to use UPnP to determine its public IP:port, then it would only get back the innermost mapping (that of the internet café's NAT) but would still have a one way voice problem. That is because the public Internet would still not recognize the IP:port that the client was giving, since a second translation occurs between the internet café's NAT and the public Internet via the ISP's NAT. There are also security issues that have not yet been addressed with UPnP.

2.2 Application Layer Gateway (ALG)

Application Layer Gateways are Routers or Firewalls including either a SIP server (with a SIP proxy and SIP registrar) that dynamically controls the firewall are currently available, or ALGS that work at a lower level than a proxy, adjusting the data packets “on-the-fly.” A common limitation of the ALG architecture is that it cannot handle secure SIP signaling via TLS (Transport Layer Security).

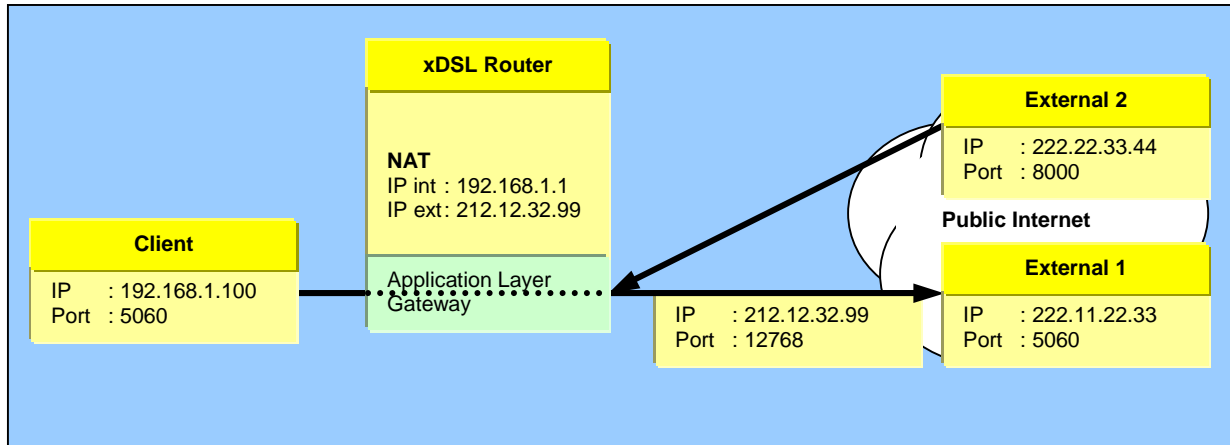


Figure 6: Application Layer Gateway (ALG)

One problem with ALGs is, that they need to be upgraded to the actual state of the SIP implementation. They furthermore modifying Protocols in a layer where shouldn't modify anything, especially because different dialects are spoken within SIP (RFC 4593, RFC 3261, 3GPP SIP, “Microsoft” SIP).

2.3 External Query

In the absence of a method of communicating with the NAT device, the next best way for a client to determine its external IP:port is to ask a server sitting outside the NAT on the public Internet how it sees the source of a packet coming from this client. In this scenario, a server sits listening for packets (call this a NAT probe). When it receives a packet, it returns a message from the same port to the source of the received packet containing the IP:port 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:port contained within the return packet is different than the IP:port that it thinks it is).
- Traversal in SIP. Which public IP:port it should use in the SDP message in order for the endpoint to reach it.

For example, if the client wants to be reached on 192.168.1.100:5060, it will first send out a query to the NAT probe from port 5060. The NAT probe will actually receive the query packet from 212.12.32.99:12768 and so it will respond to that IP:port with a packet containing 212.12.32.99:12768. The client then puts into its SDP “m= AUDIO 12768” and “c=212.12.32.99” while the client itself listens on 192.168.1.100:5060.

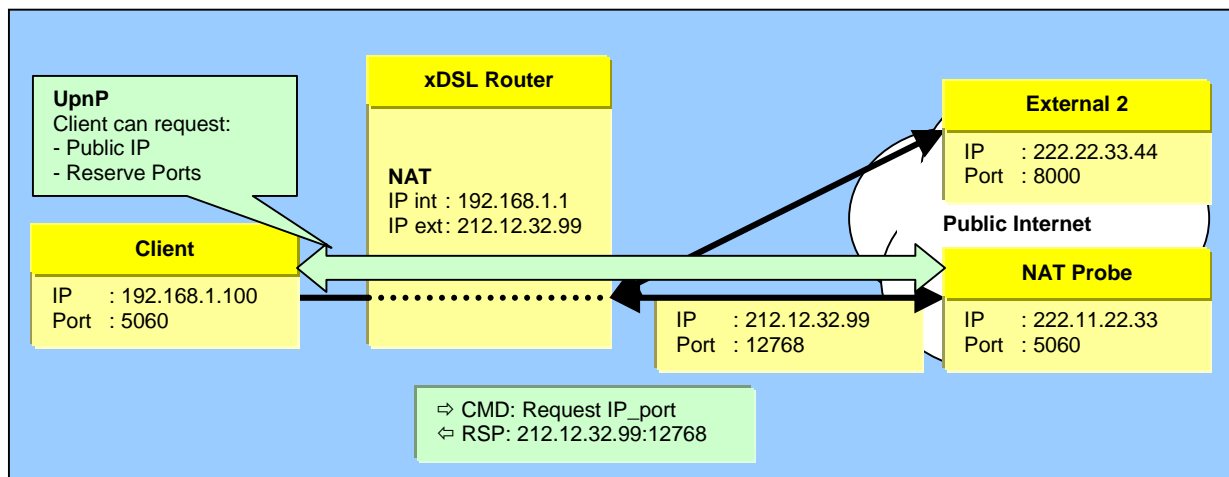


Figure 7: External Query

This will work with the following stipulations:

- The client must send and receive RTP on the same port.
- The client must send out the SIP message shortly after sending out a query to the NAT probe. If there is a long delay, the mapping may change.
- In the case of Restricted Cone or Port Restricted Cone NATs, the client must send out a packet to the endpoint before the NAT will allow packets from the endpoint through to the client. It also limits transactions to client originated (i.e. no one can contact the client whom the client did not contact first).

This will not work in the case of symmetric NATs, since the IP address of the NAT probe is different than that of the endpoint, and therefore the mapping the NAT probe sees is different than the mapping that the endpoint would use to send packets through to the client on that IP:port.

2.4 STUN

Simple Traversal of UDP Through NATs (STUN) /3/ is a protocol for setting up the kind of NAT Probe that was just described. It actually does a bit more than just return the public IP:port – it can also help determine which kind of NAT you are behind. Clients are already being developed that are STUN aware and can set their SDP messages accordingly. STUN requests specify the following parameters:

- RESPONSE-ADDRESS
- Change IP
- Change Port

The STUN server will send its response to the IP:port specified in the RESPONSE-ADDRESS attribute. If that field is not present, then the server sends its response to the IP:port that it received the request from. If both the Change IP and Change Port flags are not set, the STUN server responds from the IP:

- port that the initial packet was sent to. If the change IP flag is set, the server replies from a different IP, and if the Change Port flag is set, the server replies from a different port.

The STUN response contains the following information:

- MAPPED-ADDRESS – the IP:port of the client as seen by the first STUN server outside the NAT to receive the STUN request.
- CHANGED-ADDRESS - the IP address that would be the source of the returned response if the request had the change IP flag set.
- SOURCE-ADDRESS – the IP:port where the STUN response was sent from.

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

2.4.1 Automatic Detection of NAT Environment

Four tests are required in order for the client to determine the environment that it is situated in. The following table shows the parameters set and responses that are expected from each of these tests. Assume that there are two STUN servers available, IP1 and IP2, and they can return responses either from port 1 or port 2.

NAT	Test Destination	Change IP	Change Port	Return IP:port
Test I	IP1:1	N	N	IP1:1
Test II	IP1:1	Y	Y	IP2:2
Test III	IP2:1	N	N	IP2:1
Test IV	IP1:1	N	Y	IP1:2

Table 2: STUN Tests

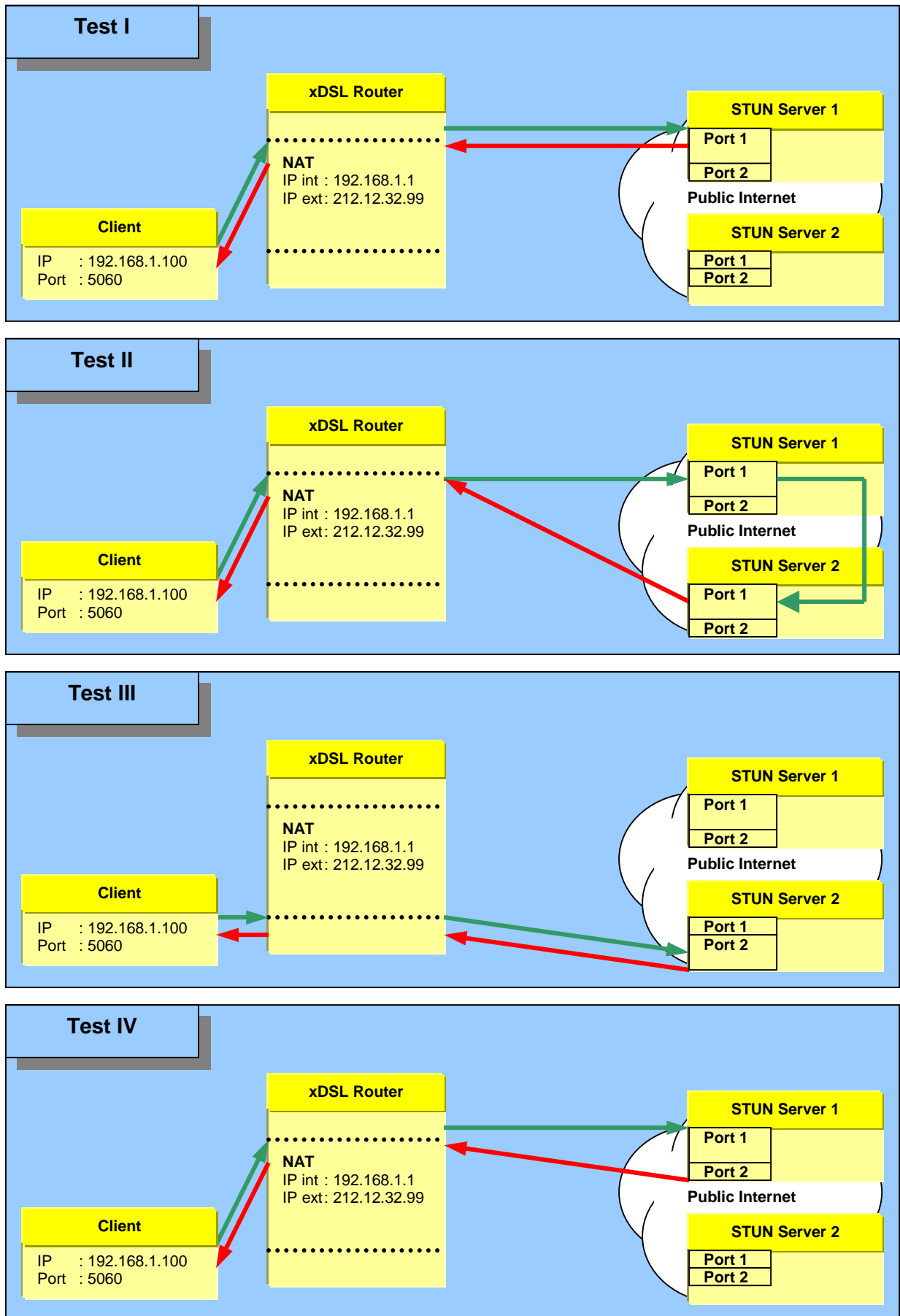


Figure 8: STUN

Traversal in SIP

In order for the client to discover its NAT environment, these four tests are run according to the following flow:

1. Test I is performed.
 - If no response is received, then the client knows it is behind a firewall that blocks UDP.
2. If a response is received, the IP address in the MAPPED-ADDRESS field of the STUN response is tested against what the client thinks its IP address is.
3. If the IP addresses match, Test II (Change IP and Port) is run.
 - If there is no response, then the client is behind a symmetric UDP firewall – that is its
 - IP address is on the open Internet, 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 a response, then the client knows that it is on the open Internet unblocked.
4. If the IP addresses in step 2 are not the same, Test II is run.
 - If the client receives a response, then it is behind a Full Cone NAT.
5. If no response is received, the client runs Test III and tests the IP address that is returned in the STUN response's MAPPED-ADDRESS field (coming from IP 2) against the MAPPED-ADDRESS that was returned in Test I (from IP 1).
 - If the two IP addresses are not the same, then the client is behind a Symmetric NAT.
6. If the two IP addresses are the same, the client runs Test IV (Change Port).
 - If a response is received, then the client is behind a Restricted NAT.
 - If no response is received, then the client is behind a Port Restricted NAT.

2.5 Connection Oriented Media (Comedia)

The above solution (NAT probe or STUN server) will only work for the first 3 types of NAT. The 4th case – symmetric NATs – will not allow this scheme since they have different mappings depending on the target IP address. So the mapping that the NAT assigns between the client and the NAT probe is different than that assigned between the client and the gateway.

In the case of a symmetric NAT, the client must send out RTP to, and receive RTP back from the same IP address. Any RTP connection between an endpoint outside a NAT and one inside a NAT must be established point-to-point, and so (even if a SIP connection has already been established) the endpoint outside the NAT must wait until it receives a packet from the client before it can know where to reply. This is known as Connection Oriented Media.

If an endpoint is meant to speak both to clients that are behind NATs and clients on the open Internet, then it must know when it can trust the SDP message that it receives in the SIP message, and when it needs to wait until it receives a packet directly from the client before it opens a channel back to the source IP:port of that packet.

One proposal /4/ for informing the endpoint to wait for the incoming packet is to add a line to the SDP message (coming from the client behind the NAT):

```
a=direction:active
```

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 RTP, and that the IP:port found in the SDP message should be ignored.

SIP clients do not currently support the ‘a=’ tag described here. Until they do, there will have to be some kind of ‘translator’ inserted into the SIP flow that can key off some other cue in order to determine that the client is behind a Symmetric NAT. Once it makes that determination, the ‘translator’ will insert the a=direction:active line into the SDP of the SIP message.

The recommended way based on RFC 3261 is to set the ‘a=’ tag (attribute field) to a=inactive, a=sendonly, a=recvonly or a=sendrecv.

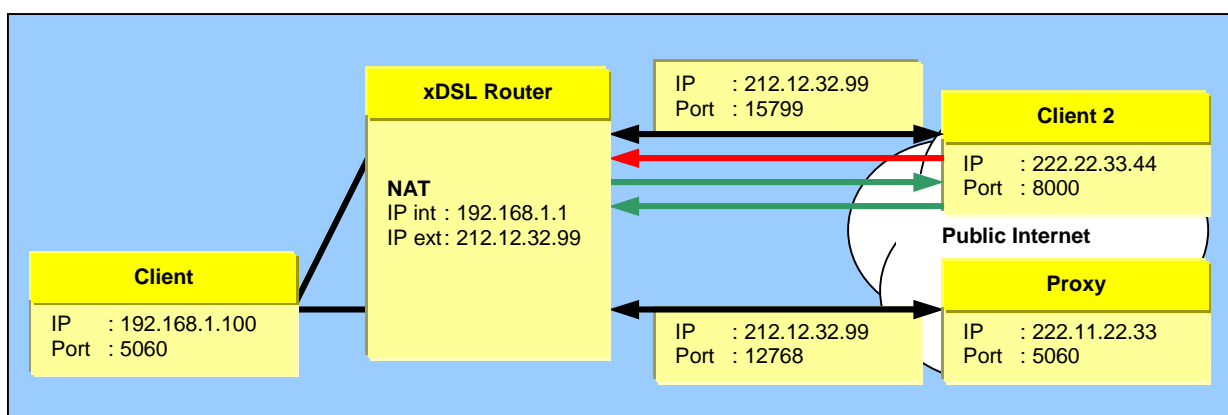


Figure 9: Connection Oriented Media (Comedia)

Unfortunately, this solution doesn't work if both Clients are behind NAT.

2.6 RTP Relay (TURN)

If an endpoint supports Connection Oriented Media, then the problem of symmetric NAT traversal is solved. Two scenarios are still problematic:

1. If the endpoint does not support the `a=direction:active` tag.
2. If both endpoints are behind Symmetric NATs

In either of these cases, one solution is to have an RTP Relay in the middle of the RTP flow between endpoints. The RTP Relay acts as the second endpoint to each of the actual endpoints that are attempting to communicate with each other. Typically, there would be a server in the middle of the SIP flow (herein called a NAT Proxy) that would manipulate the SDP in such a way as to instruct the endpoints to send RTP to the 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 uses that mapping to forward the RTP from endpoint to endpoint.

The following is a typical call flow that might be instantiated between a User Agent behind a symmetric NAT and a voice gateway on the open Internet:



Figure 10: RTP Relay

2.6.1 Call Flow

1. UA sends an INVITE to the NAT Proxy through the NAT
2. The NAT Proxy contacts the RTP Relay and requests it to set up a session.
3. The RTP Relay assigns an available pair of ports to this Call. It responds to the NAT Proxy with downstream available port in RTP Relay. The NAT Proxy uses this to modify the SDP information of the received INVITE request.
4. The NAT Proxy forwards the SIP INVITE request with modified SDP (reflecting the RTP Relay's IP:port) on to the Client 2.
5. The second client replies (in the 200 OK) with its own SDP information including the port to receive RTP packets.
6. The NAT Proxy contacts the RTP Relay to supply the IP:port of the Client 2 (if the gateway was also behind a symmetric NAT, then the NAT Proxy would instruct the Relay to wait for packets from the Client 2 before setting the IP:port to forward RTP on to the Gateway).
7. The Relay responds to the NAT Proxy with the upstream available RTP Port.
8. The NAT Proxy forwards the response upstream back to the UA after modifying the response SDP with the IP:port of the RTP Relay.
9. UA begins sending RTP to the IP:port it received in the 200 OK – to the RTP Relay.
10. RTP Relay notes the IP:port that it received the packet from (for the first packet), and passes on the packet to the IP:port of the Client 2.
11. RTP packets proceed from the Client 2 to the RTP Relay.
12. The RTP Relay forwards those packets to the client (according to IP:port that it saved when it received the first RTP packet from the client).

When BYE is received by the NAT Proxy, it forwards this information over to the RTP Relay which tears down the session.

The following considerations should be noted:

1. The client will always need to send and receive RTP on the same port.
2. This solution will work for all types of NATs, but because of the delay associated with the RTP Relay (which may be substantial, especially if the RTP Relay is not close to at least one of the endpoints), it should probably not be used unless a Symmetric NAT is involved. In other NAT scenarios, modification of the SDP will be sufficient.
3. The client will not hear any voice until the first packet is sent to the RTP Relay. That could cause problems when receiving a 183 message as part of the call setup, since the gateway at that point opens a one-way media stream and passes back network announcements over that stream. If the client has not yet sent its first RTP packet, the RTP relay does not yet know its public IP:port address.
4. This is just one way of implementing an RTP Relay. There are other possibilities, including schemes that do not insert themselves into the SIP flow.

2.7 Inbound call

Everything that has been discussed up until now - in the case of outbound calls – can be applied to inbound calling as well. Once the issue of the SIP NAT traversal has been solved, the same issues discussed above can be implemented, except in the case of symmetric NAT.

1. SDP Manipulation: If the client behind a NAT receives an INVITE, it will go out to a STUN server to find the appropriate IP:port mapping and insert that into the SDP message in the 200 OK that it returns.
2. Connection Oriented Media: The client will return the 'a=direction:active' line in the SDP of the 200 OK. In the case where this is not implemented in the client, it can use the 'c=0.0.0.0' cue in the 200 OK for a translator to pick up and insert the 'a=' line.
3. The RTP Relay and NAT Proxy will function accordingly, manipulating the SDP of the 200 OK instead of or in addition to the INVITE.

3 Analysis of different Solutions

3.1 UPnP

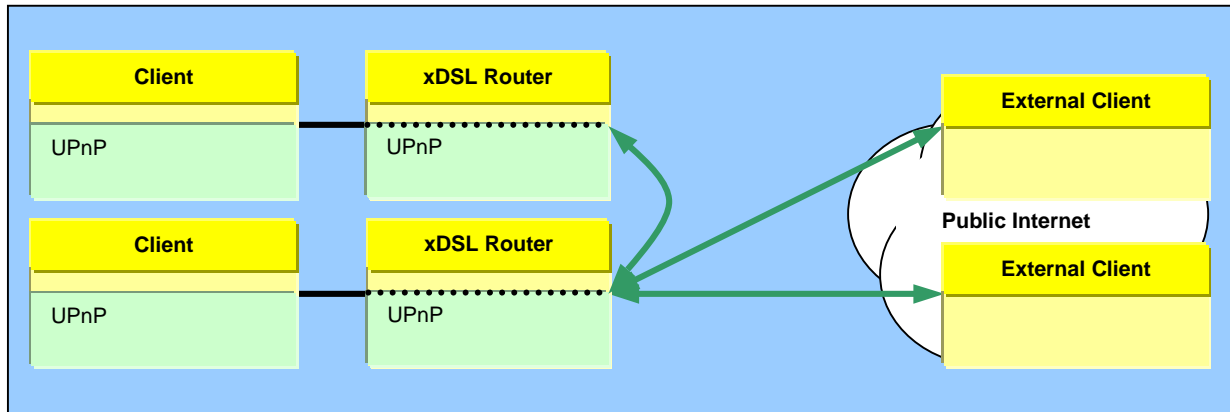


Figure 11: Involved Components for UPnP

Technique	UPnP
Advantage	<ul style="list-style-type: none"> • Works if both clients are behind NAT • Works for all 4 types of NAT • No infrastructural changes necessary • Optimizes bandwidth of the SIP network, because it'll possible to call peer-to-peer
Disadvantage	<ul style="list-style-type: none"> • Not all Routers support UPnP • Does not support cascaded private networks • High effort implementing the solution into the client • Security. Trojans can open up your router • Configuration (enabling UPnP) necessary
Business Customers	<ul style="list-style-type: none"> • No solution for Business Customers (Reason: Security)
Private Customers	<ul style="list-style-type: none"> • Good solution for residentials
Conclusion	For the residential market, UPnP is an easy – and in the long term – cheap solution.

Table 3: Analysis of UPnP

3.2 STUN (External Query)

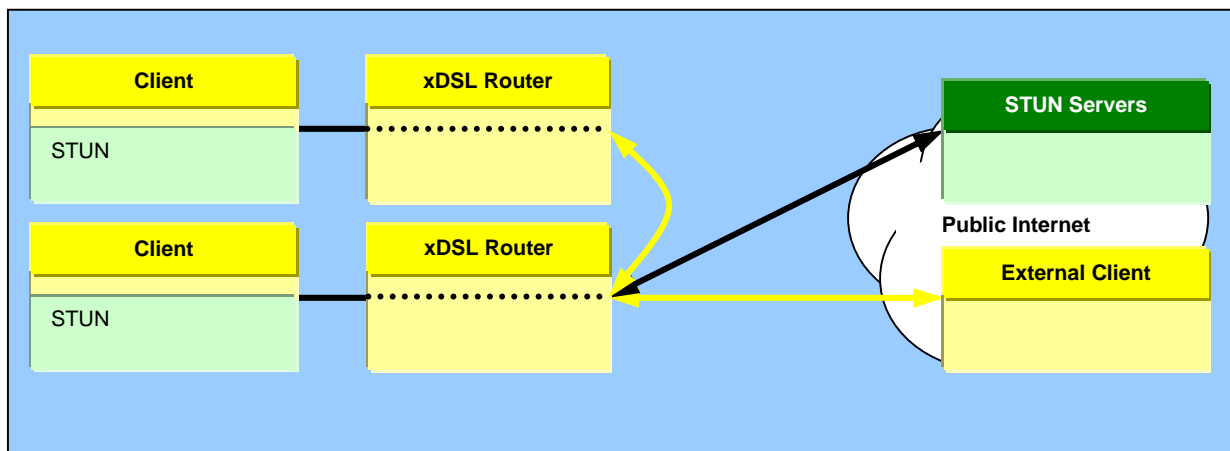


Figure 12: Involved Components for STUN

Technique	STUN
Advantage	<ul style="list-style-type: none"> • Detailed information about the NAT device • Optimizes bandwidth of the SIP network, because it'll possible to call peer-to-peer (in some cases) • No Security problem • No configuration necessary on the Router (except disabling symmetric NAT)
Disadvantage	<ul style="list-style-type: none"> • Does not work, if both clients are behind NAT • Does not support symmetric NAT (Comedia solves this problem) • Needs infrastructure (STUN Servers) • High effort implementing the solution into the client (comparable to UPnP)
Business Customers	<ul style="list-style-type: none"> • Could be useful to Business Customers, because they have little to change on the Router.
Private Customers	<ul style="list-style-type: none"> • Good solution for residentials
Conclusion	STUN is comparable to UPnP, except that it doesn't support the case if both clients are behind NAT.

Table 4: Analysis of STUN

3.3 Application Layer Gateways (ALGs)

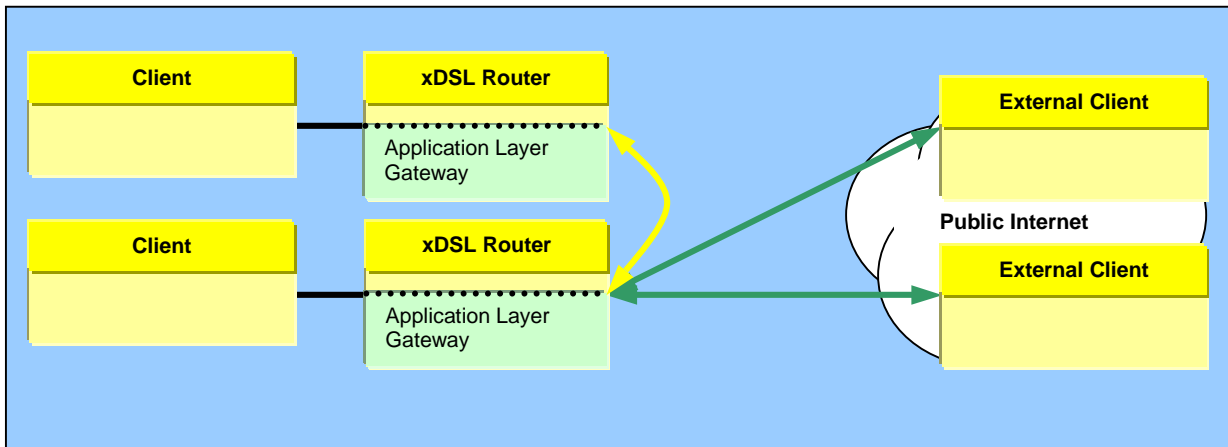


Figure 13: Involved Components for ALGs

Technique	Application Layer Gateway (ALG)
Advantage	<ul style="list-style-type: none"> No modification necessary on the Client and the Infrastructure Optimizes bandwidth of the SIP network, because it'll possible to call peer-to-peer No Security problem
Disadvantage	<ul style="list-style-type: none"> Router needs to be upgraded for new SIP features Configuration on the Router necessary. Customer needs to replace his Router with a SIP capable Router
Business Customers	<ul style="list-style-type: none"> Could be a solution for Business Customers. Might be an acceptance problem.
Private Customers	<ul style="list-style-type: none"> Solution for residentials. Acceptance to buy a new Router just to use SIP questionable.
Conclusion	Technically seen as a dirty solution. I'm not sure if customers will accept replacement of their existing Router.

Table 5: Analysis of ALGs

3.4 Connection Oriented Media (Comedia)

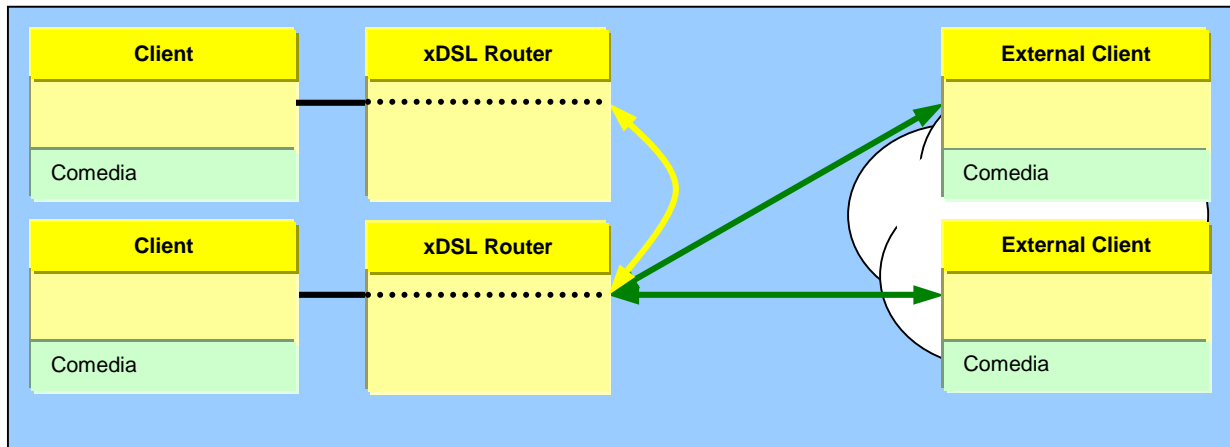


Figure 14: Involved Components for Comedia

Technique	Connection Oriented Media (Comedia)
Advantage	<ul style="list-style-type: none"> • Easy to implement • Works for Cone & Symmetric NAT, if just one subscriber is behind NAT • Good base for the other NAT Translations
Disadvantage	<ul style="list-style-type: none"> • No support if both clients are behind NAT if at least one is using Symmetric NAT. • All components need Comedia support (Proxy, Clients, Registrar, etc.)
Business Customers	<ul style="list-style-type: none"> • Solution for Business Customers. Solves not all configurations.
Private Customers	<ul style="list-style-type: none"> • Solution for residential. Solves not all configurations.
Conclusion	<p>Good solution that unfortunately doesn't solve all configurations. Good base for other enhancements, which should be implemented anyway.</p>

Table 6: Analysis of Comedia

3.5 RTP Relay (TURN)

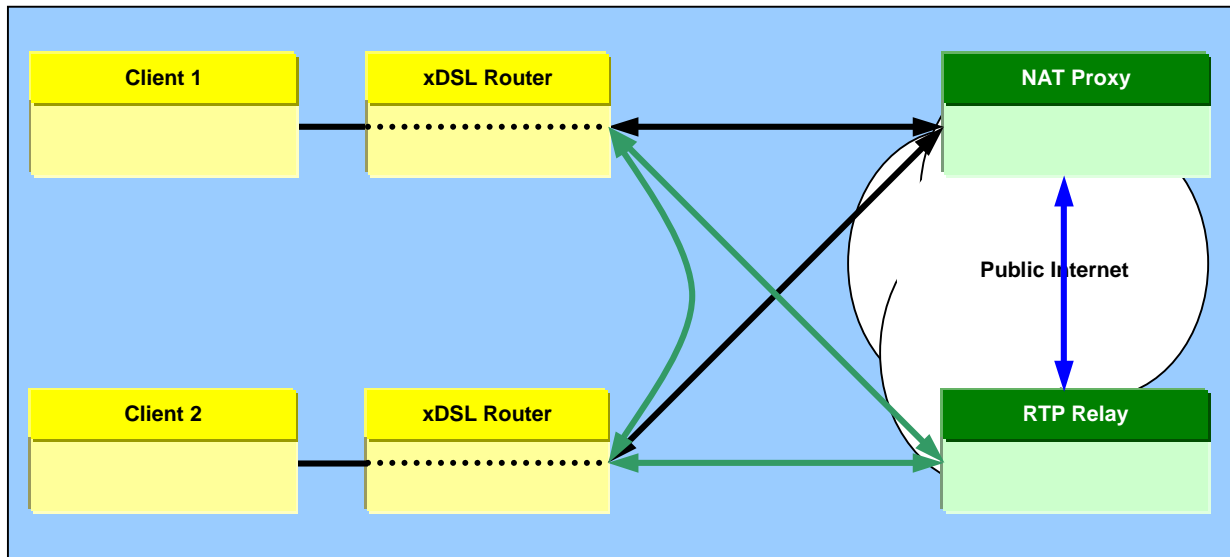


Figure 15: Involved Components for the RTP Relay(TURN)

Technique	RTP Relay (TURN)
Advantage	<ul style="list-style-type: none"> Solves all combinations (Comedia required)
Disadvantage	<ul style="list-style-type: none"> Expensive Large bandwidth consumption, because most traffic goes over the RTP Relay Single point of failure
Business Customers	<ul style="list-style-type: none"> Excellent solution for Business Customers.
Private Customers	<ul style="list-style-type: none"> Solution for residential.
Conclusion	Best but most expensive solution. It's recommended to combine this solution with other NAT translation techniques. This would allow a fallback solution for every endpoint.

Table 7: Analysis of the RTP Relay

4 Recommended architecture

This chapter describes the recommended solution for an ISP. It also contains the priority to include the different NAT translation techniques.

4.1 Architecture, 1st step: Comedia and TURN

The first step will support all customers, independent of bandwidth consumption and price of the RTP Relay. This would mean that the ISP is able to sell its solution to all customers, independent of its Internet access. This solution will always stay as a backup solution for special customers and network configurations.

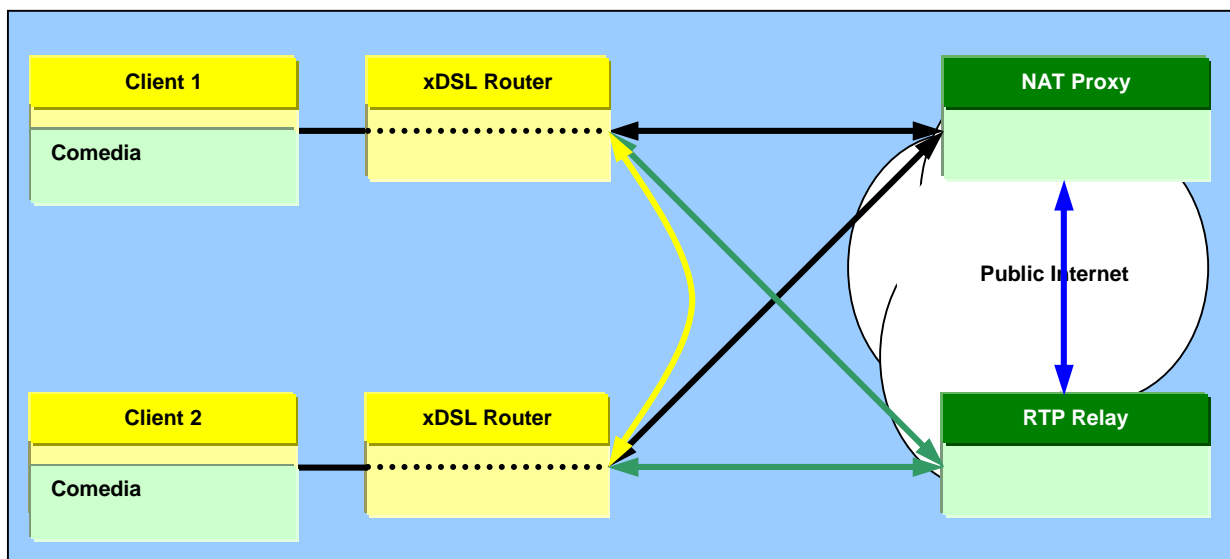


Figure 16: Architecture, 1st step: Comedia and RTP Relay

Caution:

- High bandwidth rental costs

4.2 Architecture, 2nd step: ALGs

This step can start in parallel. ALGs will help to optimize bandwidth. This solution can be sold to every customer willing to replace his Router.

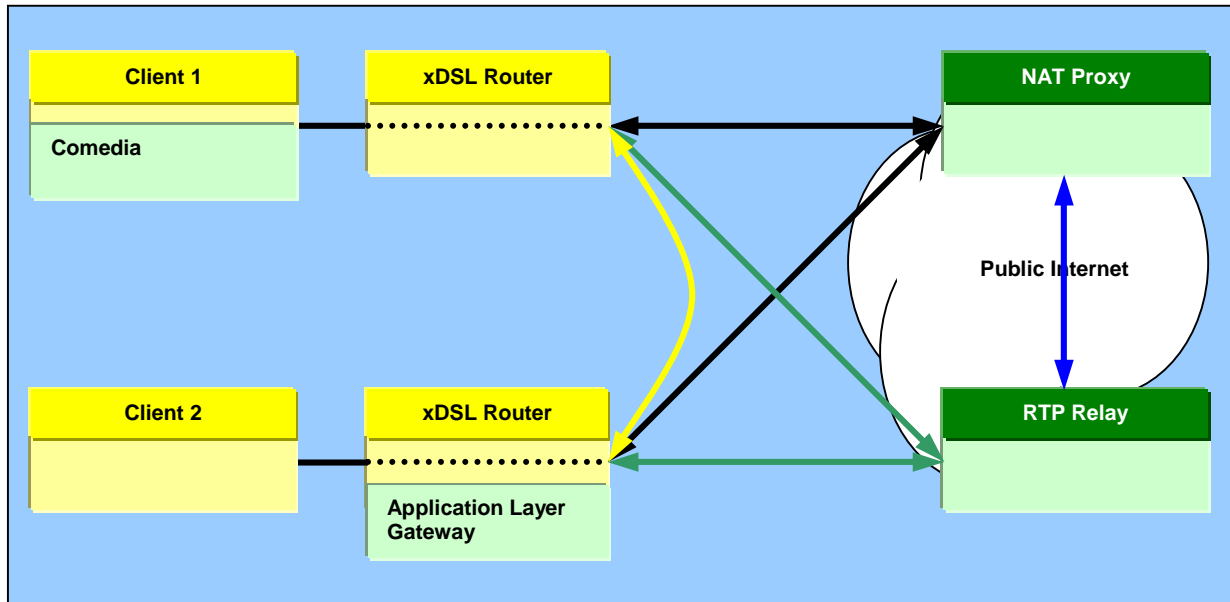


Figure 17: Architecture, 2nd step: ALGs

4.3 Architecture, 3rd step: UPnP

This step can start if there is an installed base that allows spending the effort, implementing UPnP. This solution will cover all customers, not willing to replace their expensive Routers, which already support UPnP. This will help to optimize bandwidth in a larger scale than ALGs.

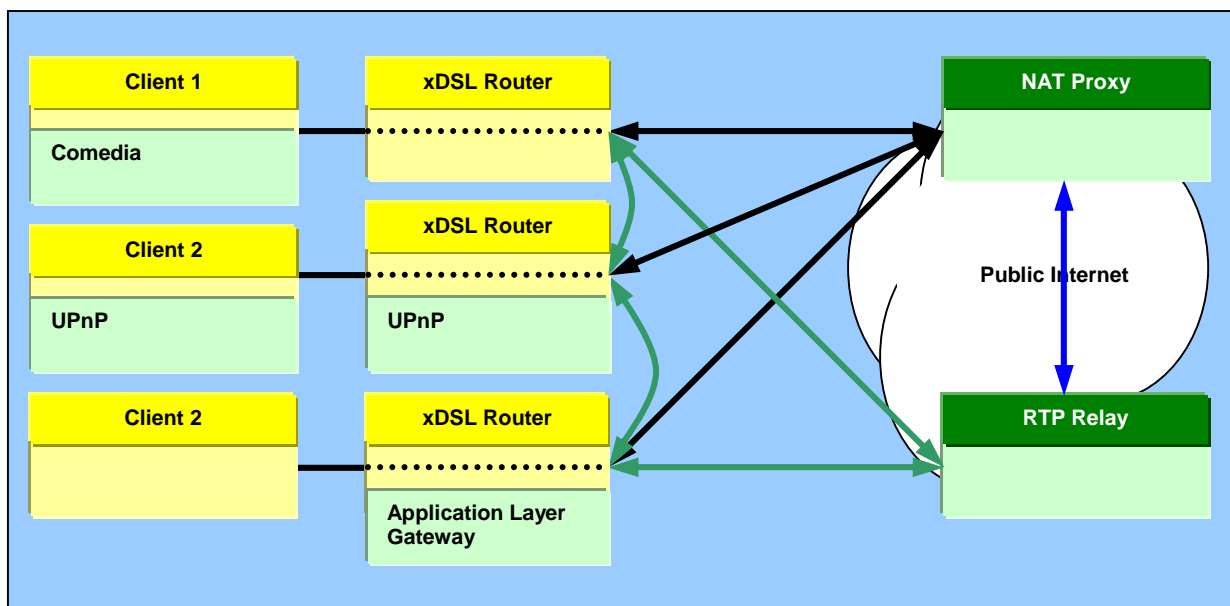


Figure 18: Architecture, 3rd step: UPnP

4.4 Architecture, 4rd step: STUN

The last step supports customers with cheap Routers that don't support UPnP.

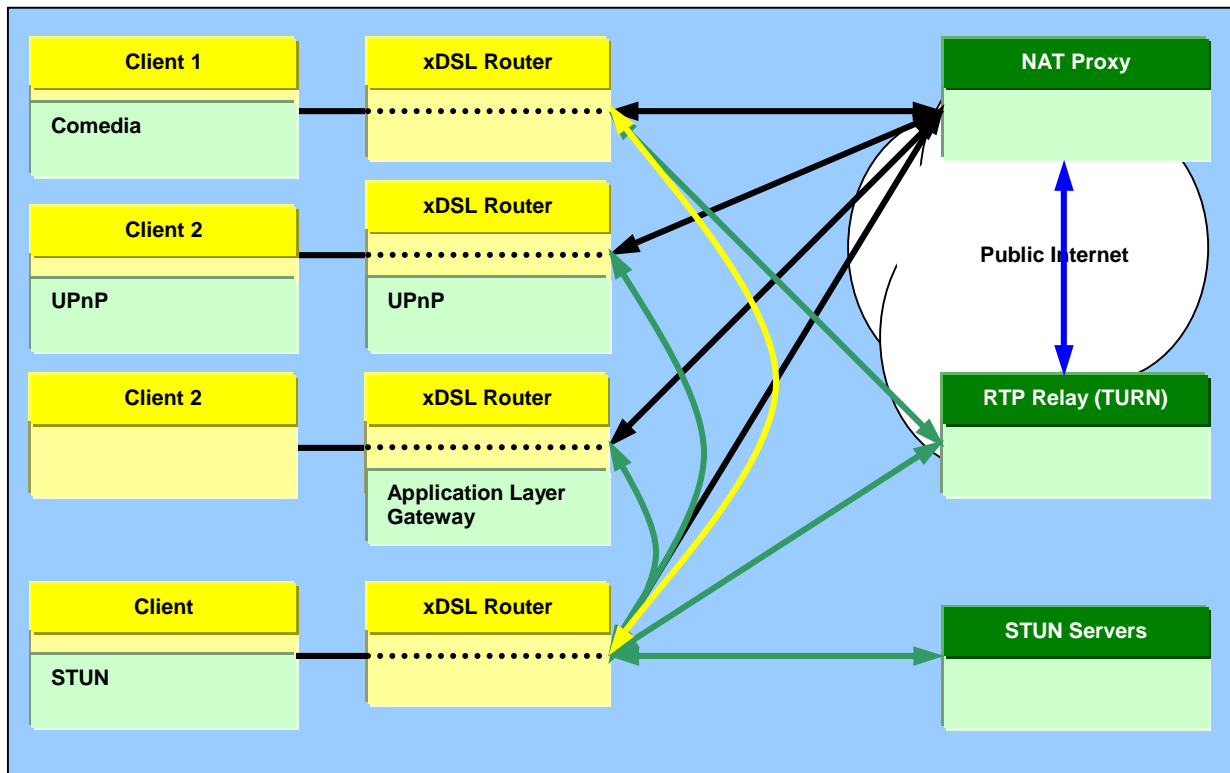


Figure 19: Architecture, 4rd step: STUN