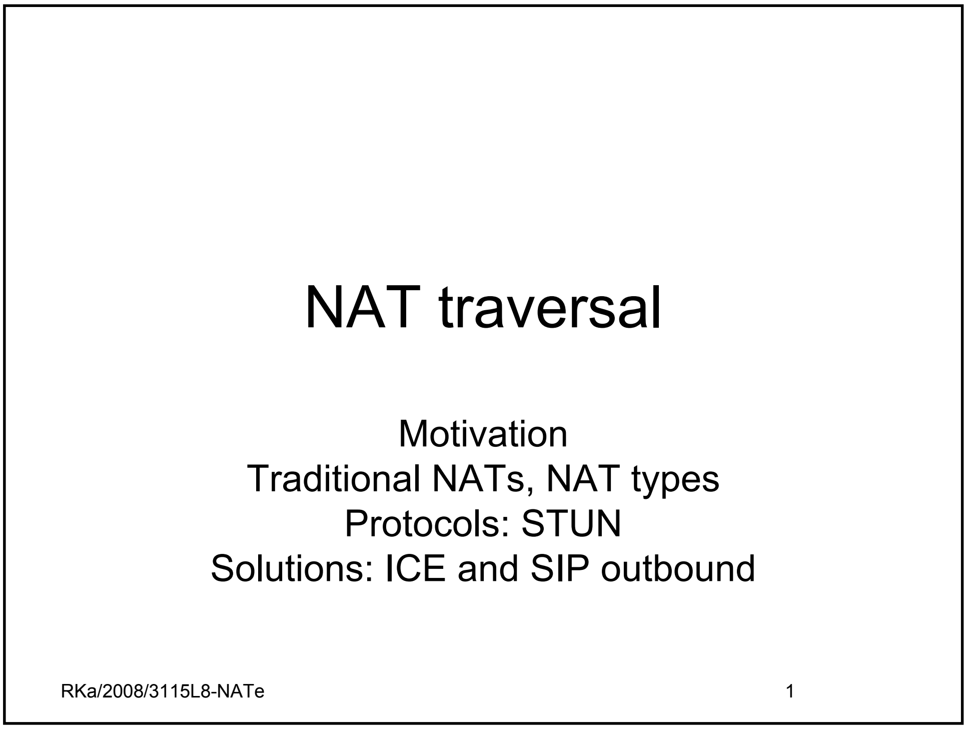
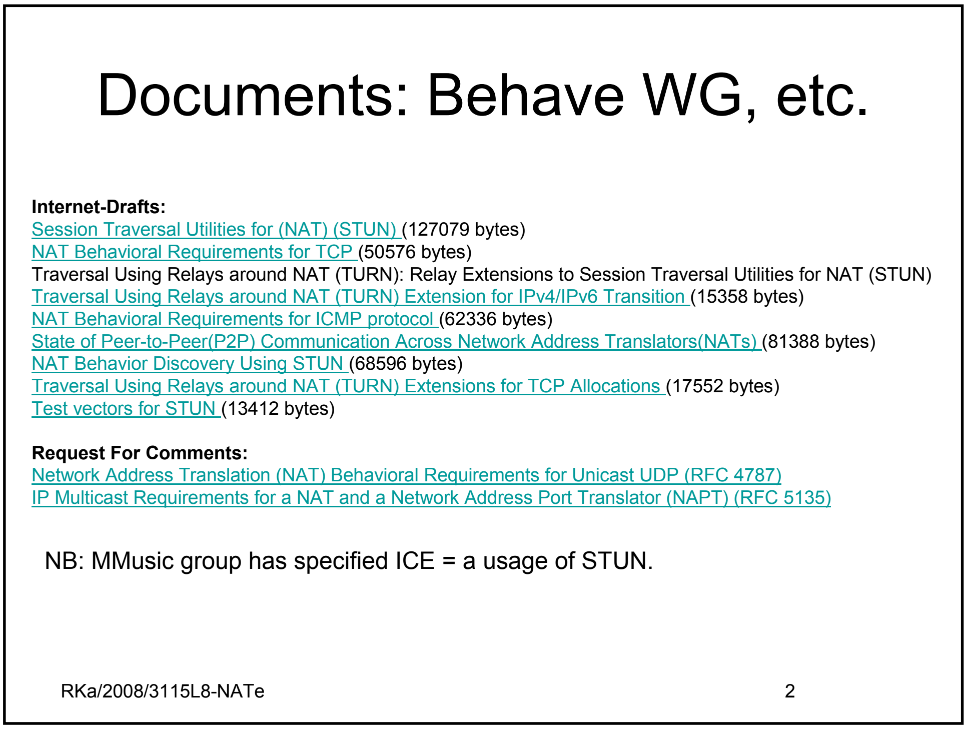


NAT traversal

Motivation
Traditional NATs, NAT types
Protocols: STUN
Solutions: ICE and SIP outbound



Documents: Behave WG, etc.

Internet-Drafts:
Session Traversal Utilities for (NAT) (STUN) (127079 bytes)
NAT Behavioral Requirements for TCP (50576 bytes)
Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)
Traversal Using Relays around NAT (TURN) Extension for IPv4/IPv6 Transition (15358 bytes)
NAT Behavioral Requirements for ICMP protocol (62336 bytes)
State of Peer-to-Peer(P2P) Communication Across Network Address Translators(NATs) (81388 bytes)
NAT Behavior Discovery Using STUN (68596 bytes)
Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations (17552 bytes)
Test vectors for STUN (13412 bytes)

Request For Comments:
Network Address Translation (NAT) Behavioral Requirements for Unicast UDP (RFC 4787)
IP Multicast Requirements for a NAT and a Network Address Port Translator (NAPT) (RFC 5135)

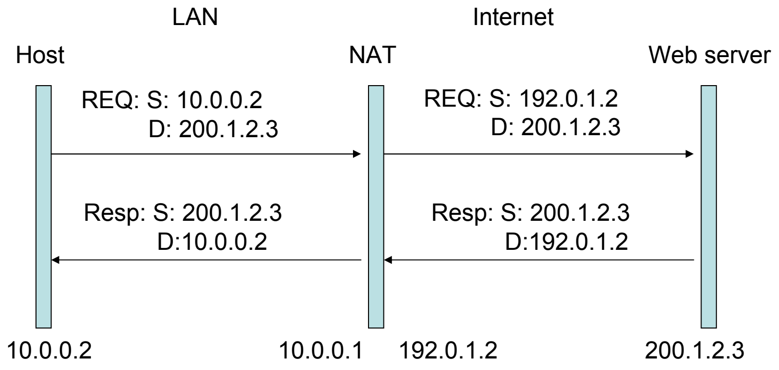
NB: MMusic group has specified ICE = a usage of STUN.

Motivation for NAT traversal

- For the purpose of IPv4 address saving, many users sit behind Network Address Translators.
  - a destination behind NAT is not reachable on the Internet
- SIP/SDP and many other protocols write IP addresses and port numbers in application level data – is unusable when message is sent through NAT
- NATs are of many types – optimal traversal depends on the NAT type

Internet is an A-subscriber’s Network! B-subscribers are not connected!

NAT were designed for web style behavior



On REQ, NAT establishes a dynamic mapping with a timeout, so that Resp can be routed back. Mapping state has at least:
host-s-address+destination address+NATs public IP address+NATs Port+timeout

About traditional NAT

RFC2663 provides information on NAT taxonomy and terminology.
Traditional NAT is the most common type of NAT device deployed.
RFC3022 describes traditional NATs. Traditional NAT has two main varieties
-- Basic NAT and Network Address/Port Translator (NAPT). NAPT is by far the most commonly deployed NAT device.

NAPT allows multiple internal hosts to share a single public IP address simultaneously. When an internal host opens an outgoing TCP or UDP session through a NAPT, the NAPT assigns the session a public IP address and port number, so that subsequent response packets from the external endpoint can be received by the NAPT, translated, and forwarded to the internal host. The effect is that the NAPT establishes a NAT session to translate the
(private IP address, private port nr) tuple → (public IP address, public port nr)tuple, and vice versa, for the duration of the session. An issue of relevance to peer-to-peer applications is how the NAT behaves when an internal host initiates multiple simultaneous sessions from a single (private IP, private port) endpoint to multiple distinct endpoints on the external network. We use, the term "NAT"
refering to both "Basic NAT" and "Network Address/Port Translator (NAPT)".

About NATs and VOIP

- Users behind a NAT use private addresses. They may e.g. get them from a DHCP server in the private network. E.g. an ADSL modem with several Ethernet ports may contain a NAT and the DHCP server. Private addresses are not unique in the Internet and can not be used for communication across the public Internet.
- When a host in the private network sends a message to the public Internet, the NAT creates a mapping:
  [priv-source IP add, source port] -> [public source IP addr, source port] +etc and will keep this mapping for a time. If within the time a packet is seen, the timeout is restarted. As a result, non-active hosts do not need to have a public IP address. When the timeout expires, the mapping is deleted. Due to a NAT, a large number of clients can use a single public IP address (how many depends on how many ports each will use simultaneously).
- In client server applications (DNS, e-mail, www etc), communication always starts from the host so NAT traversal is automatic. E.g. using DNS (a server in the public Internet), the client (even behind a NAT) can learn public IP addresses of other communicating parties such as mail server addresses. VOIP is fundamentally a peer-to-peer application, because a VOIP client must be reachable from the public Internet. Clients with private addresses are not reachable from the Internet – they must themselves take the initial step. Moreover, VOIP may send the callers IP add+port information in application messages (in signaling).

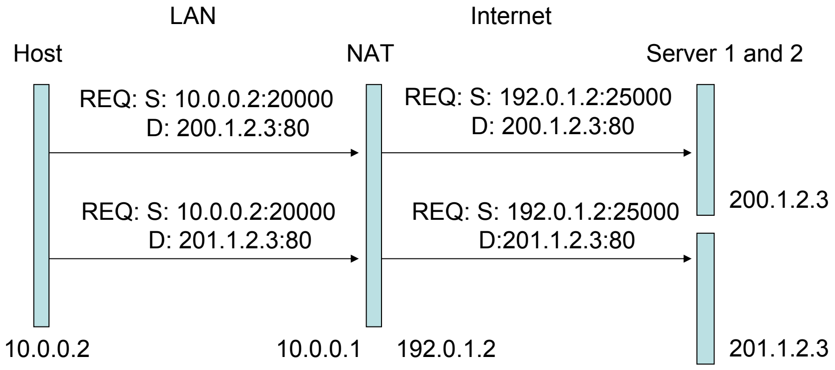
Problems created by NATs to VOIP

- Invitation (or setup message) can not be sent to a client in a private IP network, i.e. behind a NAT. This does not depend on whether the call comes from a client or a proxy in the public Internet.
  - This means that there are no B-subscribers (callees) in the Internet with NATs
- Even if the invitation goes through, sending voice packets (RTP/UDP/IP) to the B –subscriber is not possible without additional tricks, because RTP can not use the same port as signaling.
- A solution would be that ”B-subscribers” are always registered on some server in the Internet and all packets to the B-subscriber go through the server. For signaling, this might be ok (although it defiets the original purpose of NATs). For voice packets, this creates additional delay and a significant additional cost.

Classifying NAT behaviour on mapping establishment

- Mapping behavior
  - which destinations on the Internet can make use of the NAT mapping for a client
- IP Address Pooling behavior
  - A NAT may have several public IP addresses
  - How NAT uses the IP addresses as compared to using port numbers
- Port assignment in NAT

NAT mapping can be Endpoint Independent



LAN

Internet

Host

NAT

Server 1 and 2

REQ: S: 10.0.0.2:20000
D: 200.1.2.3:80

REQ: S: 192.0.1.2:25000
D: 200.1.2.3:80

REQ: S: 10.0.0.2:20000
D: 201.1.2.3:80

REQ: S: 192.0.1.2:25000
D:201.1.2.3:80

200.1.2.3

10.0.0.2

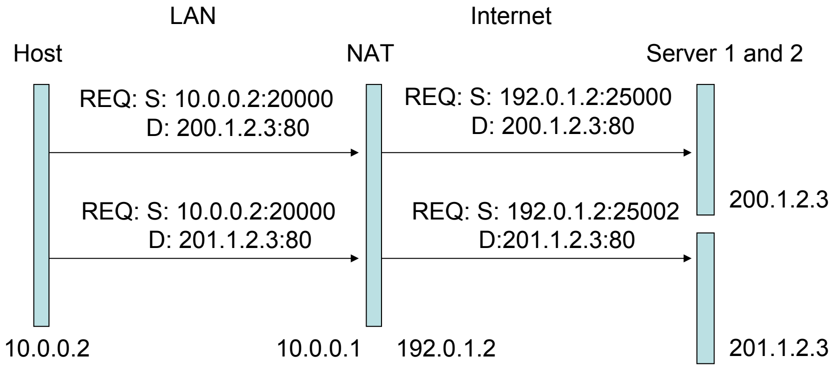
10.0.0.1

192.0.1.2

201.1.2.3

For example, the same mapping works for both VOIP signaling and media!

NAT mapping can be Address Dependent



LAN

Internet

Host

NAT

Server 1 and 2

REQ: S: 10.0.0.2:20000
D: 200.1.2.3:80

REQ: S: 192.0.1.2:25000
D: 200.1.2.3:80

REQ: S: 10.0.0.2:20000
D: 201.1.2.3:80

REQ: S: 192.0.1.2:25002
D:201.1.2.3:80

200.1.2.3

10.0.0.2

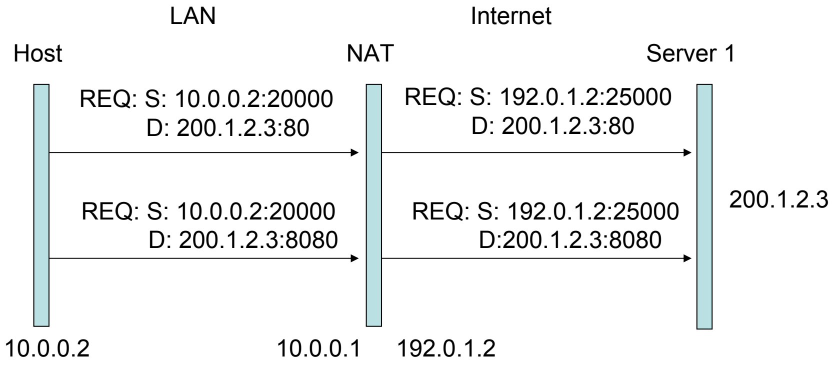
10.0.0.1

192.0.1.2

201.1.2.3

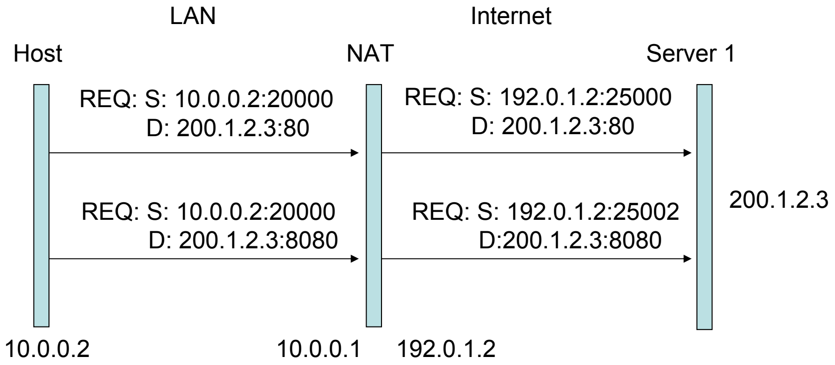
For example, different mappings are created for VOIP signaling and media!

NAT mapping can be Address Dependent (2)

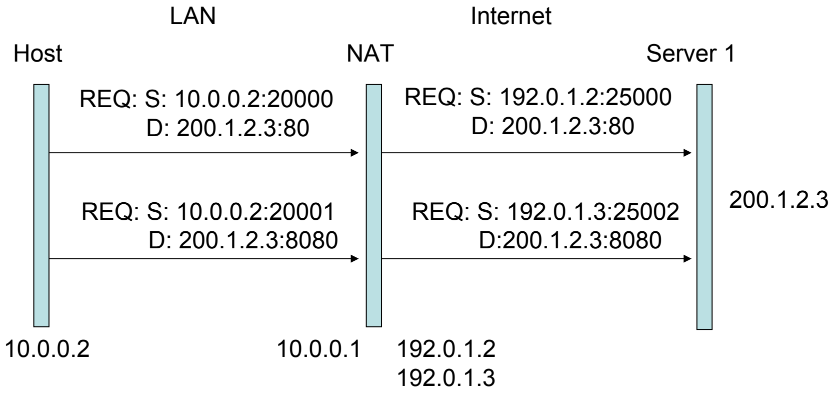


Different destination ports of the same destination address may use the same mapping in NAT.

NAT mapping can be Address and Port Dependent



NAT Address Pooling behavior: Arbitrary



LAN

Internet

Host

NAT

Server 1

REQ: S: 10.0.0.2:20000
D: 200.1.2.3:80

REQ: S: 192.0.1.2:25000
D: 200.1.2.3:80

REQ: S: 10.0.0.2:20001
D: 200.1.2.3:8080

REQ: S: 192.0.1.3:25002
D:200.1.2.3:8080

200.1.2.3

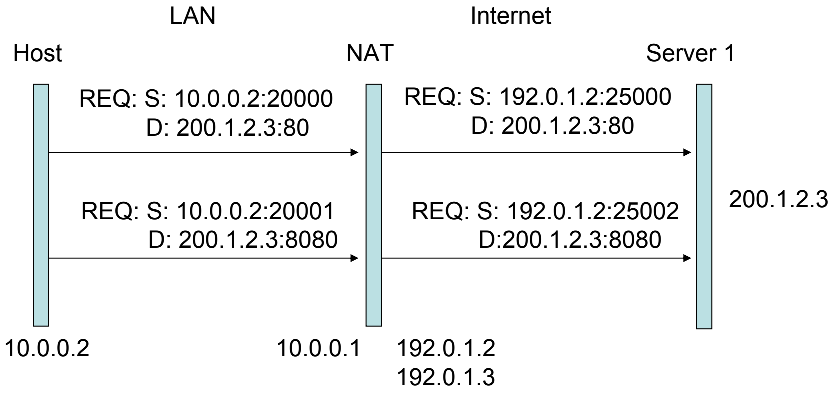
10.0.0.2

10.0.0.1

192.0.1.2
192.0.1.3

Arbitrary: an endpoint **may have** simultaneous mappings corresponding to different external IP addresses of the NAT

NAT Address Pooling behavior: Paired is recommended



LAN

Internet

Host

NAT

Server 1

REQ: S: 10.0.0.2:20000
D: 200.1.2.3:80

REQ: S: 192.0.1.2:25000
D: 200.1.2.3:80

REQ: S: 10.0.0.2:20001
D: 200.1.2.3:8080

REQ: S: 192.0.1.2:25002
D:200.1.2.3:8080

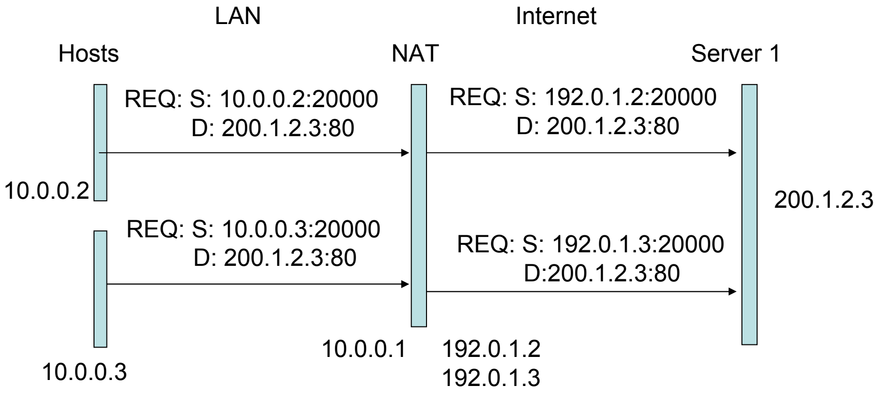
200.1.2.3

10.0.0.2

10.0.0.1

192.0.1.2
192.0.1.3

NAT mapping may or may not preserve port


LAN

Internet

Hosts

NAT

Server 1

REQ: S: 10.0.0.2:20000
D: 200.1.2.3:80

REQ: S: 192.0.1.2:20000
D: 200.1.2.3:80

10.0.0.2

200.1.2.3

REQ: S: 10.0.0.3:20000
D: 200.1.2.3:80

REQ: S: 192.0.1.3:20000
D:200.1.2.3:80

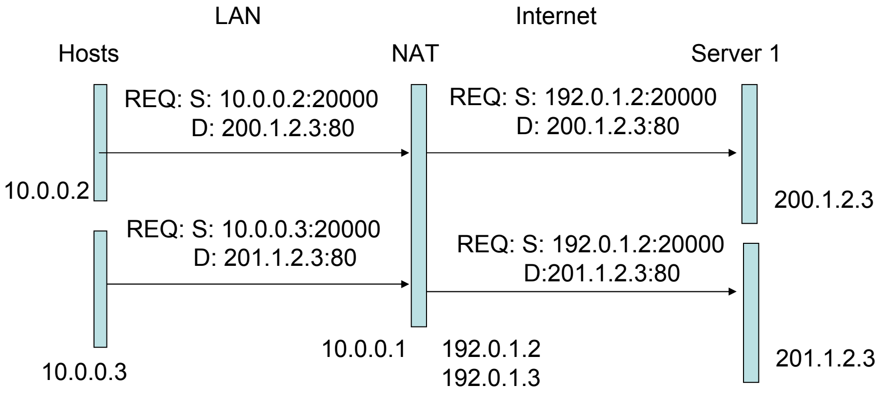
10.0.0.1

192.0.1.2
192.0.1.3

10.0.0.3

Port preservation preserves the port as long as there are available IP addresses in the NAT’s pool. When addresses are all used, NAT may no longer try to preserve port

Port Overloading preserves port and routes responses on source address


LAN

Internet

Hosts

NAT

Server 1

REQ: S: 10.0.0.2:20000
D: 200.1.2.3:80

REQ: S: 192.0.1.2:20000
D: 200.1.2.3:80

10.0.0.2

200.1.2.3

REQ: S: 10.0.0.3:20000
D: 201.1.2.3:80

REQ: S: 192.0.1.2:20000
D:201.1.2.3:80

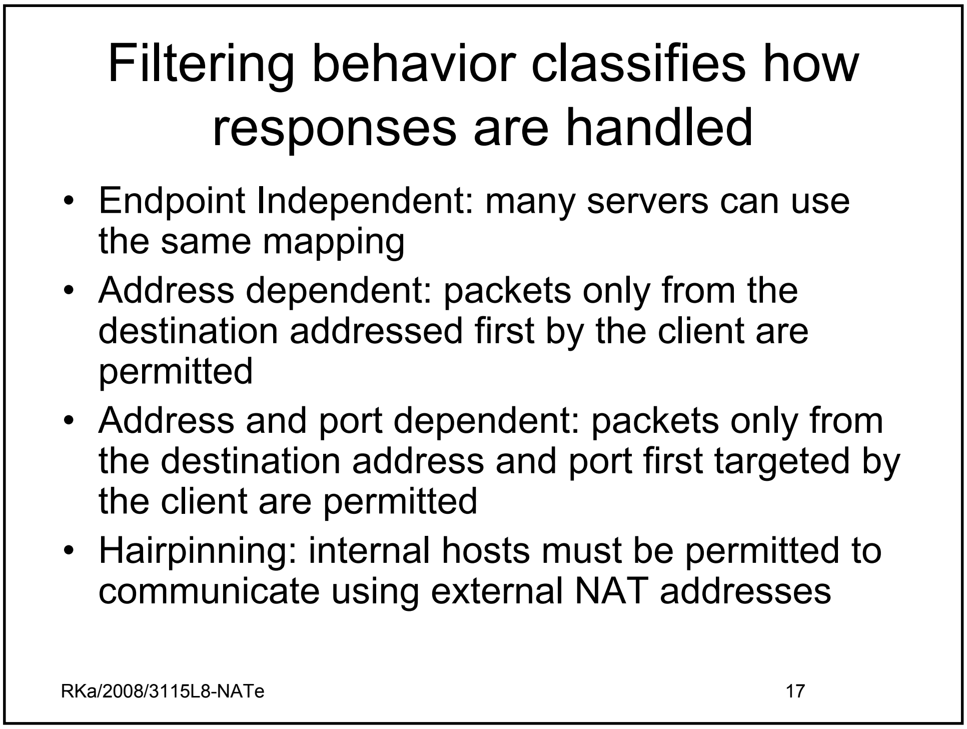
10.0.0.1

192.0.1.2
192.0.1.3

201.1.2.3

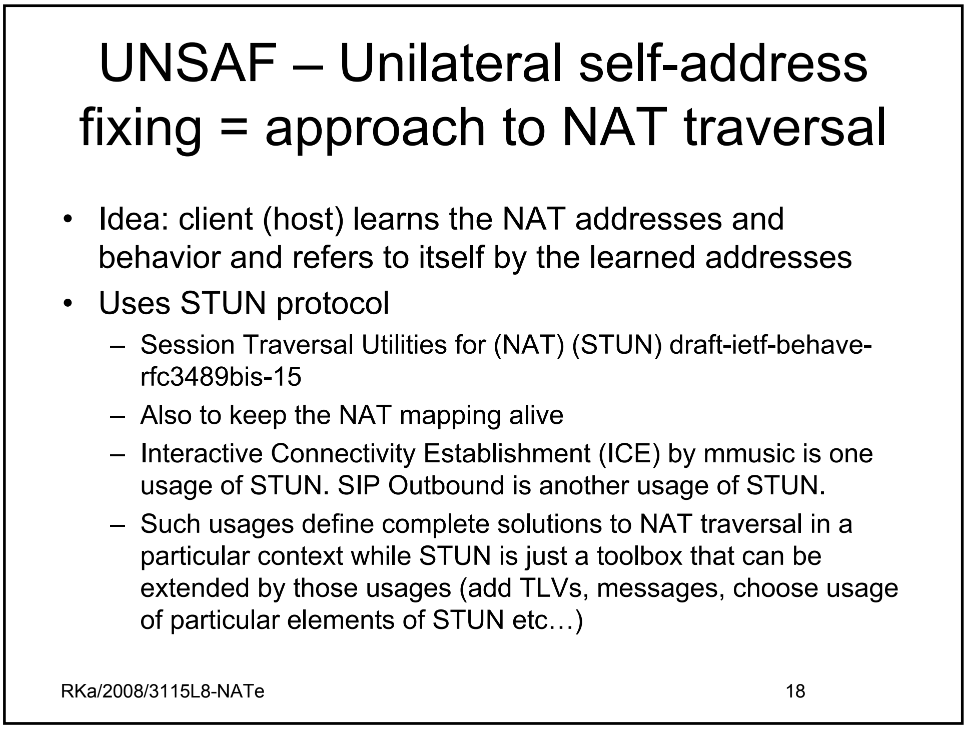
10.0.0.3

This is NOT RECOMMENDED!



Filtering behavior classifies how responses are handled

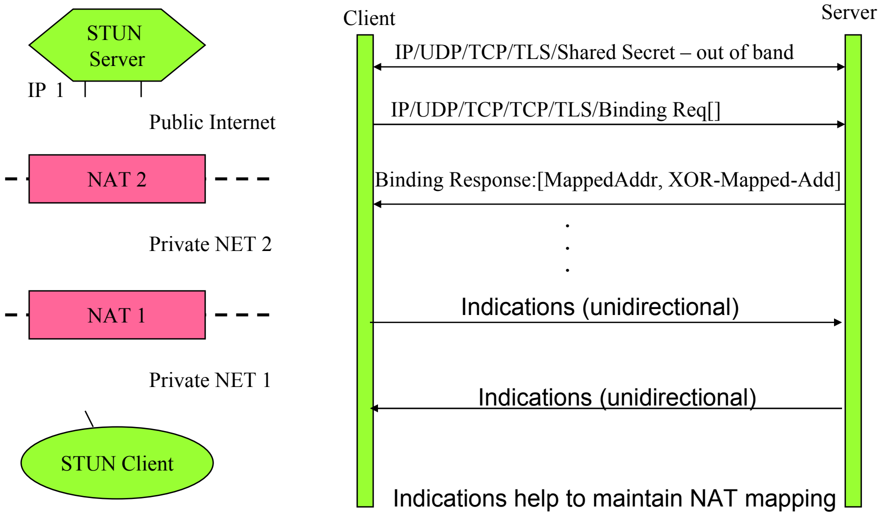
- Endpoint Independent: many servers can use the same mapping
- Address dependent: packets only from the destination addressed first by the client are permitted
- Address and port dependent: packets only from the destination address and port first targeted by the client are permitted
- Hairpinning: internal hosts must be permitted to communicate using external NAT addresses



UNSAF – Unilateral self-address fixing = approach to NAT traversal

- Idea: client (host) learns the NAT addresses and behavior and refers to itself by the learned addresses
- Uses STUN protocol
  - Session Traversal Utilities for (NAT) (STUN) draft-ietf-behave-rfc3489bis-15
  - Also to keep the NAT mapping alive
  - Interactive Connectivity Establishment (ICE) by mmusic is one usage of STUN. SIP Outbound is another usage of STUN.
  - Such usages define complete solutions to NAT traversal in a particular context while STUN is just a toolbox that can be extended by those usages (add TLVs, messages, choose usage of particular elements of STUN etc…)

STUN model assumes nested NATs


STUN Server
IP 1
Public Internet
NAT 2
Private NET 2
NAT 1
Private NET 1
STUN Client
Client
Server
IP/UDP/TCP/TLS/Shared Secret – out of band
IP/UDP/TCP/TCP/TLS/Binding Req[]
Binding Response:[MappedAddr, XOR-Mapped-Add]
Indications (unidirectional)
Indications (unidirectional)

Indications help to maintain NAT mapping

Types of NAT can be discovered by sending responses from different source address and port

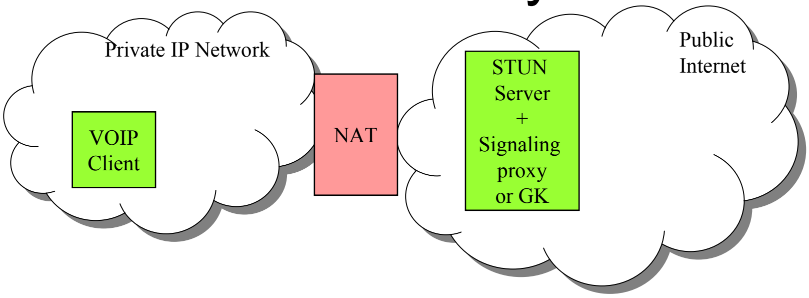
This is not part of STUN itself.

STUN plays with the identity of the user: opens a door for inpersonation. Therefore, security, excluding man-in-the-middle attacks is crucial!

To make best of use of STUN, STUN messages may need to be multiplexed with an application protocol (i.e. use the same port as the application).

When a SIP application fills in SDP fields and some SIP fields, NAT traversal needs to be taken into account!

Use of STUN by VOIP

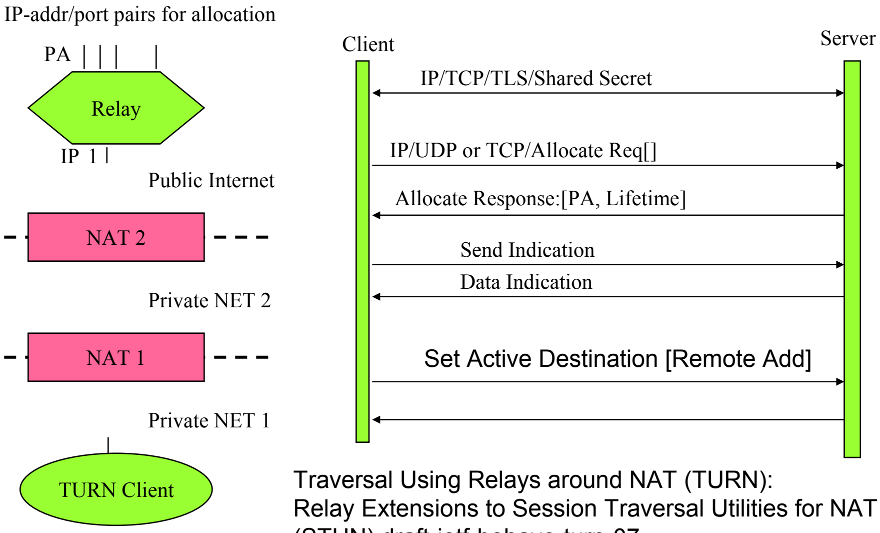


- VOIP client establishes a connection to the STUN server and learns the type of NAT + the address mapping the NAT creates for the client.
- In case of SIP signaling, the client registers the learned public address in SIP registrar/proxy.
- Now the client can be reached with signaling and invited to a session. Even voice over RTP (that comes possibly from a different IP address than signaling) can go to voip client, but only in case the NAT gives an endpoint independent mapping or address dependent mapping. In case of address and port dependent mapping, client must first send RTP before it can receive, but this is not a big deal.

Relay operations in STUN (TURN)

- Allocate request / response
  – Allocate an external (public) address at the relay
  – Responses carry a MAPPED-ADDRESS
- Send indication
  – Send data to a remote endpoint through the relay in a STUN wrapper
- Data indication
  – Data received from remote endpoints through the relay in a STUN wrapper
- Set Active Destination request / response
  – Send and receive data to and from a single remote endpoint without using Send and Data wrappers
- Connect request / response
  – Requests the relay to establish a TCP connection with the remote endpoint
- Connection Status Indication
  – The relay informs the endpoint about the status of a TCP connection with the remote endpoint
  – LISTEN, ESTABISHED, CLOSED

Relay model extends STUN



Traversal Using Relays around NAT (TURN):
Relay Extensions to Session Traversal Utilities for NAT
(STUN) draft-ietf-behave-turn-07

Interactive Connectivity Establishment- ICE is used by IMS for NAT traversal by media flows

- A usage of STUN with Relay capability
- Tries to minimize the use of relays for media.

Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols draft-ietf-mmusic-ice-19 (ca 120 pages!):

ICE works by including a multiplicity of IP addresses and ports in SDP offers and answers, which are then tested for connectivity by peer-to-peer connectivity checks. The IP addresses and ports included in the SDP.
The connectivity checks are performed using STUN.
ICE also makes use of Traversal Using Relay NAT (TURN).
ICE allows for address selection for multi-homed and dual-stack (IPv4/6) hosts.
ICE is not intended for NAT traversal for SIP, which is assumed to be provided via another mechanism [I-D.ietf-sip-outbound]

ICE collects candidate addresses

- Possible candidates are:
  - own IP address,
  - reflexive IP address (oubound NAT address)
  - and TURN address
- Both communicating parties collect candidates and test address pairs with STUN in a predetermined order to find the minimal config for communication
- STUN checks are multiplexed with RTP/RTCP

NAT traversal by SIP

- Managing Client Initiated Connections in the Session Initiation Protocol (SIP),
  draft-ietf-sip-outbound-13 (March 21st, 2008)
- SIP User Agent (on user's device) can maintain several registrations in the network for robustness – if one fails, another remains and user stays reachable
  - This time STUN needs to be multiplexed with SIP
  - STUN may need to be multiplexed with compressed SIP (i.e. SigComp
  - It is recommended that keep-alives are sent every 120s.
- Inbound session traffic (incoming call signaling) will reuse an existing ”flow” that is maintained alive thru NAT using TCP SYNs or STUN
- Does not need TURN because SIP assumes a registrar and proxy in the network thru which reachability is maintained. It is sufficient to learn the NAT outbound IP address and Port to fill in SIP/SDP data

Summary of NAT traversal

- NATs have been poorly specified and they try to keep invisible to applications. NAT behaviour was described by IETF after they were widely deployed – description had to be rewritten at least once. With the new description, some recommendations were given on what to avoid in NATs
- STUN and TURN provide a toolbox for NAT traversal. Several WGs have described complete solutions for NAT traversal on top of STUN/TURN – ICE and SIP-outbound are relevant to signaling. Both are still in draft stage.
  - one can claim that at least partially these solutions defeat the reason why NATs were invented – i.e. address space preservation. However, by making better use of the port space, the result is that de-facto addressing is extended with port numbering.
- Beyond extending the scalability of IPv4 address space NATs serve also security needs – attacks to user hosts behind NATs are much more difficult than to hosts that have public IP addresses. For this reason, requirements that NATs should be present in IPv6 networks as well are quite popular