Peer-to-Peer Systems and Security

Network Address Translation

Christian Grothoff

Technische Universität München

April 8, 2013

"Freedom of connection with any application to any party is the fundamental social basis of the Internet. And now, is the basis of the society built on the Internet." –Tim Berners-Lee

Network Address Translation

Why NAT?

- IPv4 address shortage
- “private” network / firewall

Network Address Translation

Why is NAT relevant for P2P networks?

- common type of **middlebox**
- Many variations in the specific implementation
- IP violation creates issues for TCP, UDP, ICMP, SCTP, ...
- Problems: classification, detection, traversal, application impact

Network Address Translation

Why is NAT relevant for P2P networks?

- common type of **middlebox**
- Many variations in the specific implementation
- IP violation creates issues for TCP, UDP, ICMP, SCTP, ...
- Problems: classification, detection, traversal, application impact

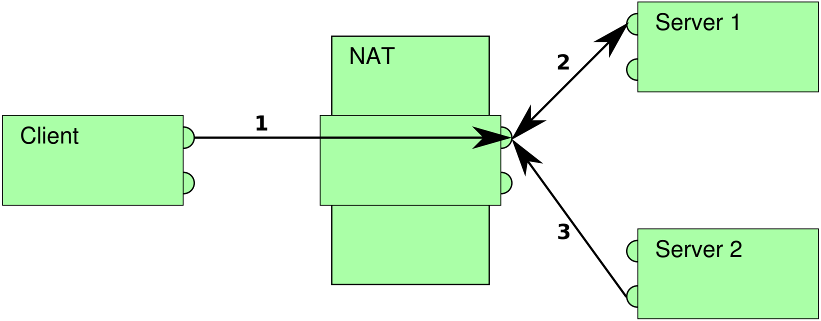
⇒ Thousands of papers (Google Scholar gives $\geq$ 1 million results for “network address translation”)

Network Address Translation: Classification

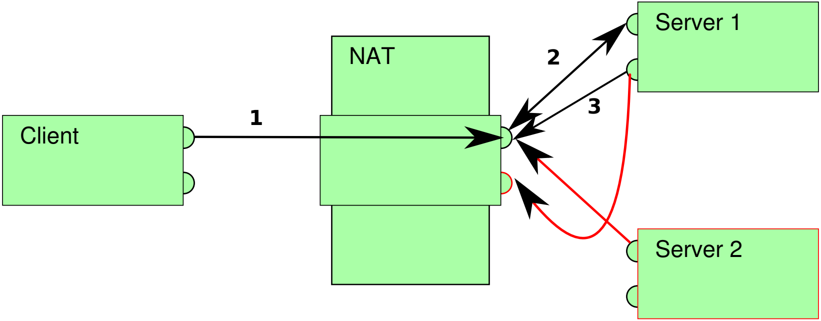
A **well-known** classification scheme uses:

- Full-cone NAT
- (Address)-restricted-cone NAT
- Port-restricted cone NAT
- Symmetric NAT

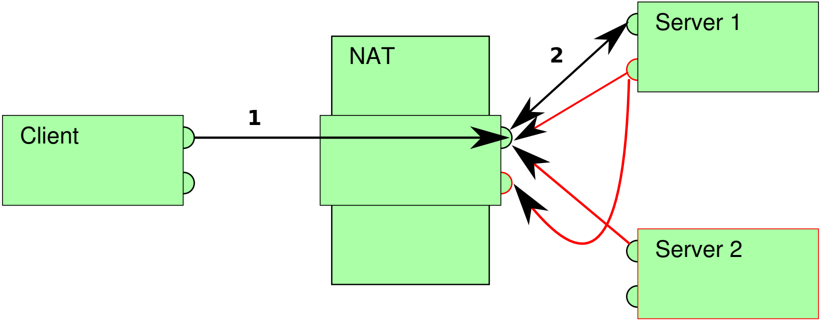
Full-Cone NAT



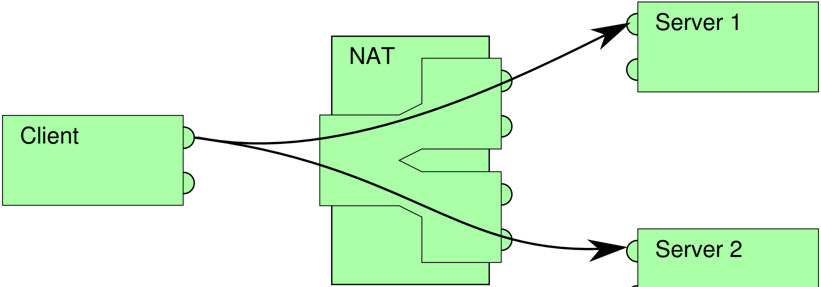
Address-restricted NAT



Port- and Address-restricted NAT



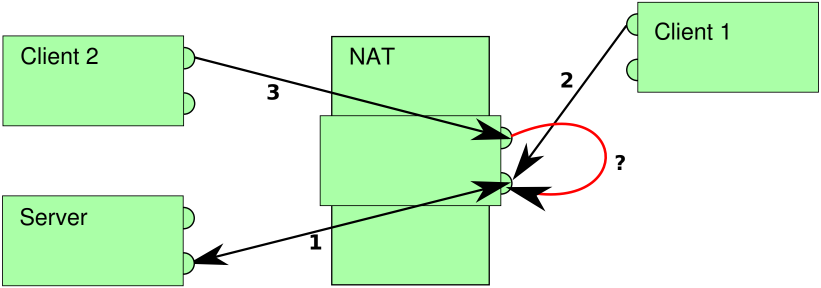
Symmetric NAT



General Properties for NAT

- Which fields have to match for the NAT to map a packet from the outside?
- How long do mappings last?
- Does the NAT track the TCP session state?
- How does the NAT select the external port? (port preservation, linear, random)
- What happens with multiple inside devices using the same source port?
- How are errors (TCP RST, ICMP) processed?
- Which protocols (UDP, TCP, ICMP, SCTP, IPSec) are supported?

NAT hair-pinning



NAT Protocol Translation (NAT-PT)

NAT-PT can be used to translate from IPv4 to IPv6 (or vice versa),

NAT Protocol Translation (NAT-PT)

NAT-PT can be used to translate from IPv4 to IPv6 (or vice versa), but ...

- IP options missmatch
- ICMP code missmatch
- DNS return values need to be translated (DNS-ALG)

Problems with NAT

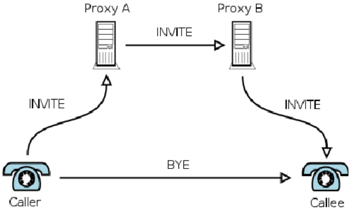
NAT breaks the end-to-end principle!

- Global IP unknown to local software
- Incoming connections “not possible”
- Protocols may include (LAN) addresses in WAN traffic payload

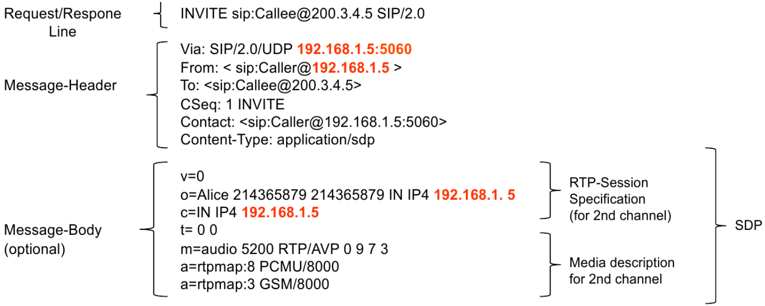
⇒ Communication becomes difficult

⇒ Particularly bad if both peers are behind NAT

Example: SIP



SIP Payload



Request/Respone Line

```
INVITE sip:Callee@200.3.4.5 SIP/2.0
```

Message-Header

```
Via: SIP/2.0/UDP 192.168.1.5:5060
From: < sip:Caller@192.168.1.5 >
To: <sip:Callee@200.3.4.5>
CSeq: 1 INVITE
Contact: <sip:Caller@192.168.1.5:5060>
Content-Type: application/sdp
```

Message-Body (optional)

```
v=0
o=Alice 214365879 214365879 IN IP4 192.168.1. 5
c=IN IP4 192.168.1.5
t= 0 0
m=audio 5200 RTP/AVP 0 9 7 3
a=rtpmap:8 PCMU/8000
a=rtpmap:3 GSM/8000
```

RTP-Session Specification (for 2nd channel)

Media description for 2nd channel

SDP

Simple solutions

- NATed peer initiates connection (Gnutella's PUSH)
- Require "super peers" to not be behind NAT, limit all-to-all communication to super peers
- Use common ports (80, 443) to get past firewall rules

NAT Traversal [5]

- Explicit support by the NAT (Static port forwarding, UPnP, NAT-PMP, ALG)
- NAT-behaviour based approaches (Hole punching, STUN [4])
- External data-relay (TURN [1])
- Autonomous NAT Traversal [2]

DNAT / PMP / UPnP

- DNAT / port forwarding allows inbound connections

DNAT / PMP / UPnP

- DNAT / port forwarding allows inbound connections
- Port Mapping Protocol (PMP) allows LAN applications to request DNAT entries and discover external IP

DNAT / PMP / UPnP

- DNAT / port forwarding allows inbound connections
- Port Mapping Protocol (PMP) allows LAN applications to request DNAT entries and discover external IP
- UPnP is an **insane** protocol, that (among other things) also allows applications to request DNAT entries and determine external IP
- Both usually fail for cascaded NATs, and are often disabled for security reasons

Application Layer Gateway (ALG)

- implemented by NAT
- common protocols that include addresses: SIP, FTP, IRC
- NAT may or may not support it!

Application Layer Gateway (ALG)

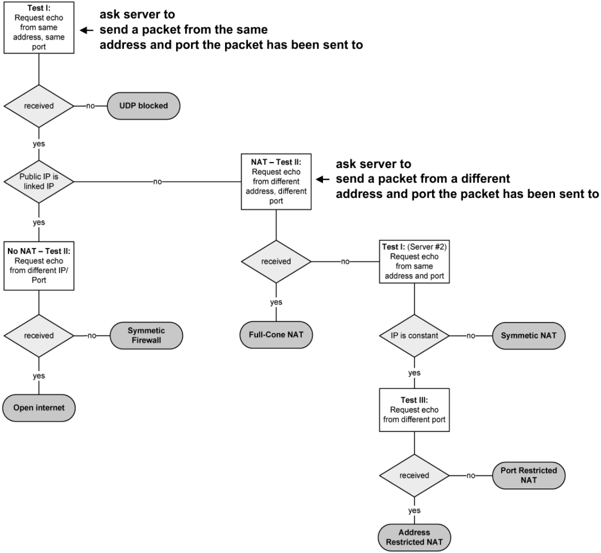
- implemented by NAT
- common protocols that include addresses: SIP, FTP, IRC
- NAT may or may not support it!

Idea: use “pretend” FTP to open port for non-FTP applications! [6]

Session Traversal Utilities for NAT (STUN)

- Determines external transport address (IP + port)
- Lightweight client-server protocol on top of UDP
- Algorithm to discover NAT type (server needs 2 public IPs)
- STUN server can also act as rendezvous point

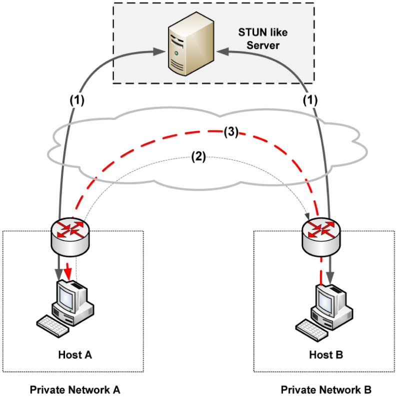
STUN Algorithm


Test I: Request echo from same address, same port
received
no
UDP blocked
yes
Public IP is linked IP
no
NAT – Test II: Request echo from different address, different port
yes
No NAT – Test II: Request echo from different IP/ Port
received
no
Symmetic Firewall
yes
Open internet
received
yes
Full-Cone NAT
no
Test I: (Server #2) Request echo from same address and port
IP is constant
no
Symmetic NAT
yes
Test III: Request echo from different port
received
no
Port Restricted NAT
yes
Address Restricted NAT

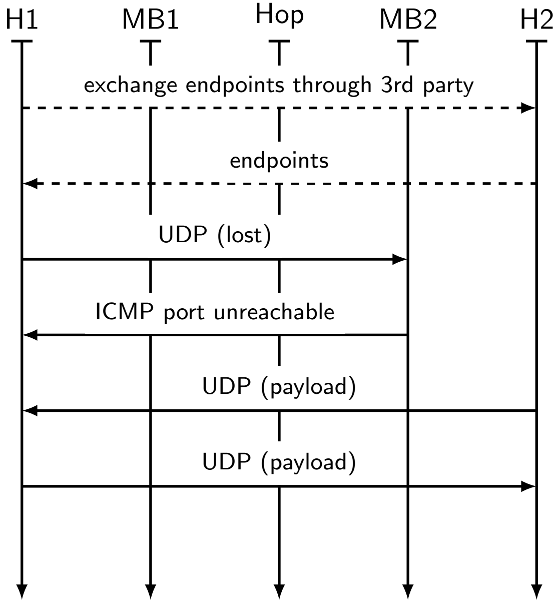
ask server to
send a packet from the same
address and port the packet has been sent to

ask server to
send a packet from a different
address and port the packet has been sent to

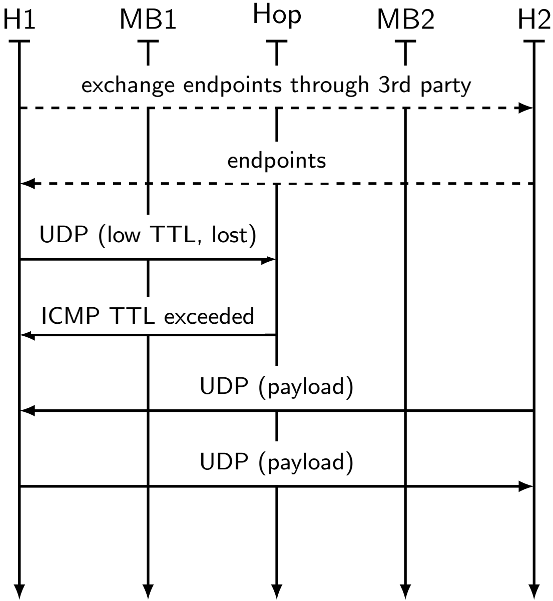
STUN for Rendezvous and Hole Punching



UDP hole punching with ICMP Unreachable



UDP hole punching with ICMP TTL exceeded



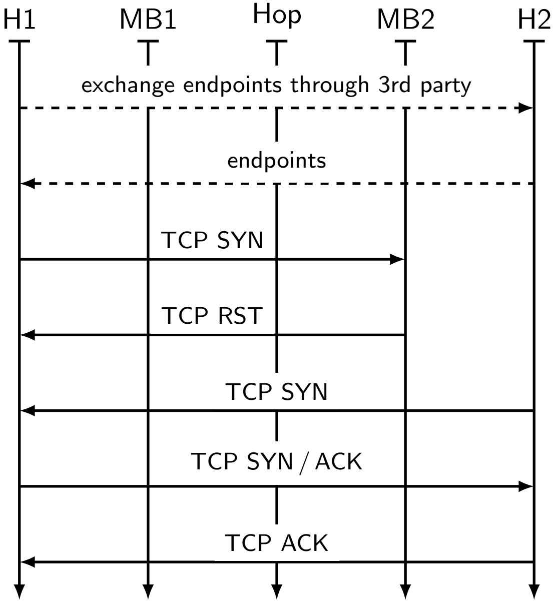
Example: UDP-based traversal

Given a NAT that preserves ports for UDP with external address “natIP”, this can suffice:

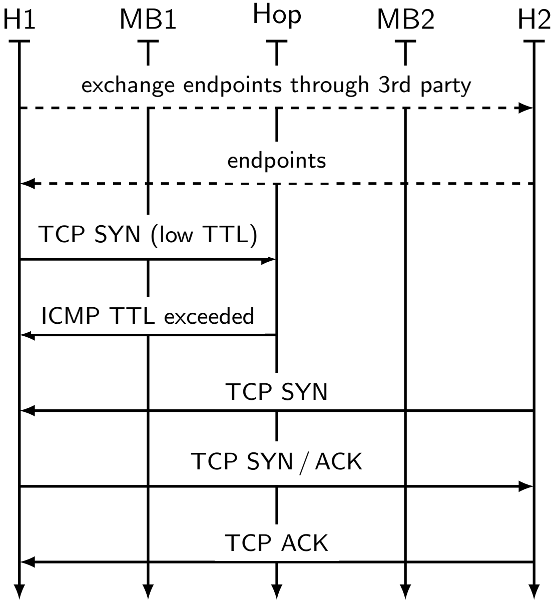
```
nat/1 $ nc -u -l -p 20000
client$ watch echo "hello" | nc -p 5000 -u natIP 20000 -w 1
nat/2 $ echo hello | nc -p 20001 -u clientIP 5000
```

If UDP hole punching is used, how could a STUN server launch a MitM attack?

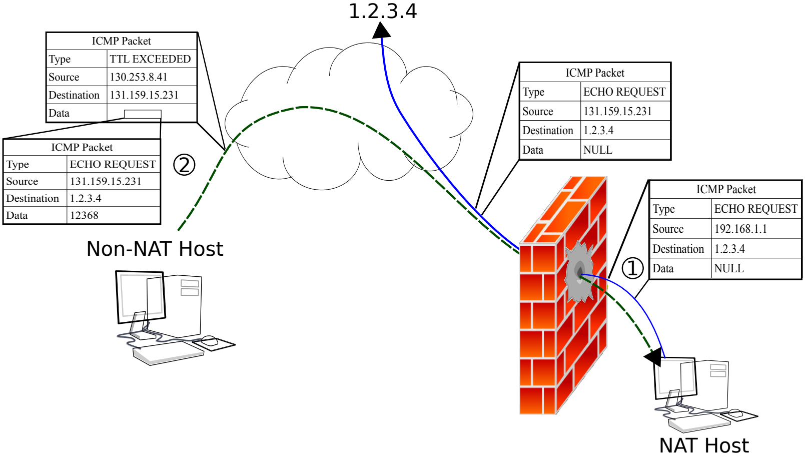
TCP hole punching with RST



TCP hole punching with ICMP TTL



Autonomous NAT Traversal



ICMP Packet	
Type	TTL EXCEEDED
Source	130.253.8.41
Destination	131.159.15.231
Data	

ICMP Packet	
Type	ECHO REQUEST
Source	131.159.15.231
Destination	1.2.3.4
Data	12368

ICMP Packet	
Type	ECHO REQUEST
Source	131.159.15.231
Destination	1.2.3.4
Data	NULL

ICMP Packet	
Type	ECHO REQUEST
Source	192.168.1.1
Destination	1.2.3.4
Data	NULL

Autonomous NAT Traversal: discussion

Enables initiation of connections to hosts behind NAT without involving a third party at the time.

+ Simpler to implement
+ Efficient, completely distributed method
+ Third party can not observe connections
+ Works well for UDP and TCP
- Does not work as often as techniques involving 3rd party

Using UDP instead of ICMP ECHO REQUEST

+ No RAW sockets required for sending periodic requests
+ Might help punch hole
- Slightly bigger messages
- Smaller response payload (32 bits only)
- May fail if NAT remaps ports

NAT Traversal Summary

There are many non-trivial methods for NAT traversal:

- Explicit support by the NAT (Static port forwarding, UPnP, NAT-PMP, ALG)
- NAT-behaviour based approaches (Hole punching, STUN)
- External data-relay (TURN)
- Autonomous NAT Traversal

None of these is perfect, NAT traversal usually uses a combination of techniques (see ICE [3]).

Network Neutrality

Phone system design:

- Quality-of-service for voice (provisioned bandwidth)
- Payment models:
  - charges based on source and destination
    (country, mobile / landline / service numbers)
  - caller-pays

Network Neutrality

Phone system design:

- Quality-of-service for voice (provisioned bandwidth)
- Payment models:
  - charges based on source and destination (country, mobile / landline / service numbers)
  - caller-pays
  - callee-pays

Network Neutrality

Phone system design:

- Quality-of-service for voice (provisioned bandwidth)
- Payment models:
  - charges based on source and destination (country, mobile / landline / service numbers)
  - caller-pays
  - callee-pays
  - everybody-pays: {roaming-, per-call-, monthly service-, directory listing-, phone number-, caller-ID-, billing-...} charges

Network Neutrality

Phone system design:

- Quality-of-service for voice (provisioned bandwidth)
- Payment models:
  - charges based on source and destination (country, mobile / landline / service numbers)
  - caller-pays
  - callee-pays
  - everybody-pays: {roaming-, per-call-, monthly service-, directory listing-, phone number-, caller-ID-, billing-...} charges

Internet design:

- "best-effort" IP forwarding, agnostic to source, destination and payload

Network Neutrality

Phone system design:

- Quality-of-service for voice (provisioned bandwidth)
- Payment models:
  - charges based on source and destination (country, mobile / landline / service numbers)
  - caller-pays
  - callee-pays
  - everybody-pays: {roaming-, per-call-, monthly service-, directory listing-, phone number-, caller-ID-, billing-...} charges

Internet design:

- "best-effort" IP forwarding, agnostic to source, destination and payload
- Payment model? DARPA!

Paradoxes in Phones Systems

- Alice can call Bob for free (flatrate), but Bob pays to call Alice ("call me back, please?")

Paradoxes in Phones Systems

- Alice can call Bob for free (flatrate), but Bob pays to call Alice (“call me back, please?”)
- Calling mobile phones in the US costs the callee, but calling mobile phones in Europe costs the caller

Paradoxes in Phones Systems

- Alice can call Bob for free (flatrate), but Bob pays to call Alice ("call me back, please?")
- Calling mobile phones in the US costs the callee, but calling mobile phones in Europe costs the caller

Phone companies charge where they can, not where it makes technical sense!

Ideas for Internet Surcharges

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic
- Using Voice-over-IP

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic
- Using Voice-over-IP
- Downloading videos

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic
- Using Voice-over-IP
- Downloading videos
- Accessing Google (advanced search) in addition to Bing (basic search)

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic
- Using Voice-over-IP
- Downloading videos
- Accessing Google (advanced search) in addition to Bing (basic search)
- Accessing non-European service providers

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic
- Using Voice-over-IP
- Downloading videos
- Accessing Google (advanced search) in addition to Bing (basic search)
- Accessing non-European service providers
- Using UDP

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic
- Using Voice-over-IP
- Downloading videos
- Accessing Google (advanced search) in addition to Bing (basic search)
- Accessing non-European service providers
- Using UDP
- Privacy (not selling your connection data)

Ideas for Internet Surcharges

- Global IP address (IPv4, IPv6)
- Incoming connections
- non-HTTP traffic
- Using Voice-over-IP
- Downloading videos
- Accessing Google (advanced search) in addition to Bing (basic search)
- Accessing non-European service providers
- Using UDP
- Privacy (not selling your connection data)

Have you read your ISP's terms of service?

Have you read your ISP's terms of service?

Would your friends understand such restrictions?

Have you read your ISP's terms of service?

Would your friends understand such restrictions?

Would enough of them care to switch providers?

Have you read your ISP's terms of service?

Would your friends understand such restrictions?

Would enough of them care to switch providers?

Can they switch providers?

Have you read your ISP's terms of service?

Would your friends understand such restrictions?

Would enough of them care to switch providers?

Can they switch providers?

Should business models be regulated?

Creative Methods

Creative Methods

- Sell “low-latency” plan to service providers

⇒ Micosoft pays for 50 ms Bing, Google gets 5s penalty latency

Creative Methods

- Sell “low-latency” plan to service providers

⇒ Micosoft pays for 50 ms Bing, Google gets 5s penalty latency

- Give customers illusion of speed

⇒ Prefer traffic to URLs with “speedtest” in them

Creative Methods

- Sell “low-latency” plan to service providers

⇒ Micosoft pays for 50 ms Bing, Google gets 5s penalty latency

- Give customers illusion of speed

⇒ Prefer traffic to URLs with “speedtest” in them

- Reduce quality of VoiP calls via artifical drops

⇒ Force customers to pay extra for voice service

Creative Methods

- Sell “low-latency” plan to service providers

⇒ Micosoft pays for 50 ms Bing, Google gets 5s penalty latency

- Give customers illusion of speed

⇒ Prefer traffic to URLs with “speedtest” in them

- Reduce quality of VoiP calls via artifical drops

⇒ Force customers to pay extra for voice service

- Reduce bandwidth for P2P traffic

⇒ Entice users to pay for services

Can we detect such creative methods?

Can we detect such creative methods?

How can we circumvent them?

Can we detect such creative methods?

How can we circumvent them?

Can enough of society understand the problem?

Can we detect such creative methods?

How can we circumvent them?

Can enough of society understand the problem?

Will society establish laws to ward against this?

Questions?

?

“Just as we are beginning to see the power that free resources produce, changes in the architecture of the Internet–both legal and technical–are sapping the Internet of this power. Fueled by bias in favor of control, pushed by those whose financial interest favor control, our social and political institutions are ratifying changes in the Internet that will reestablish control, in turn, reduce innovation on the Internet and in society generally.” –Lawrence Lessig

References

R. Mahy, P. Matthews, and J. Rosenberg.

Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN).

RFC 5766 (Proposed Standard), April 2010.

Andreas Müller, Nathan Evans, Christian Grothoff, and Samy Kamkar.

Autonomous nat traversal.

In *10th IEEE International Conference on Peer-to-Peer Computing (IEEE P2P 2010)*, pages 61–64, 2010.

J. Rosenberg.

Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols.

RFC 5245 (Proposed Standard), April 2010.

Updated by RFC 6336.

J. Rosenberg, R. Mahy, P. Matthews, and D. Wing.

Session Traversal Utilities for NAT (STUN).

RFC 5389 (Proposed Standard), October 2008.

P. Srisuresh, B. Ford, and D. Kegel.

State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs).

RFC 5128 (Informational), March 2008.

M. Wander, S. Holzapfel, A. Wacker, and T. Weis.

Ntalg - tcp nat traversal with application-level gateways.

In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 46–47, 2012.