SIP-assisted NAT Traversal

Jianping Pan
web.uvic.ca/~pan
October 25, 2005
Internet was ...

- infrastructures
  - routers, end-hosts
- applications
  - 1-way data transfer
- requirements
  - data integrity

- global addressable and end-to-end reachable
"Internet" now ...

- many mid-boxes
- firewalls, NATs
- many applications
  - 2 or more-way
  - many requirements
  - security!

- SIP
- ~SS7?
- IPsec
- IPv4/6
- SSL/TLS
  - over TCP

- signaling
- media

SIP UA
Proxy
NAT
IP phone
SIP UA
firewall
Firewalls and NATs

• firewalls and NATs usually work hand-to-hand
• firewalls: packet filtering w/ (known) rules

-NATs: initially as a quick-fix to IPv4 address shortage
-now pervasive in every networking scenario
• translate source/destination address/port
• update other related information (checksum etc.)

-NATs: initially as a quick-fix to IPv4 address shortage
-now pervasive in every networking scenario
• translate source/destination address/port
• update other related information (checksum etc.)
IPsec and NAT

- authenticated
- encrypted+authed

- upper-layer header inaccessible
- IP header cannot be modified

- transport mode
- tunnel mode

- UDP-encapsulated IPsec NAT traversal
- MTU discovery?
SIP and NAT

• session establishment for UAs behind NATs
• information embedded in SDP (local vs. global view)
• NAT traversal is still an open problem!

A

local addr/port

B

local addr/port

AAA/radius

• invite

• invite

register/auth/invite

global addr/port

• media?

media?
Roadmap

• Introduction
  – why IPsec, SIP, and NAT cannot work together
• Why NAT traversal is so difficult?
• NAT traversal approaches
• SIP-assisted NAT traversal
  – with *ordinary* applications and NATs and IPsec
• Network reliability & security in a big picture
Types of NATs: full cone

• NAT behaviors were never regulated!

• outgoing mapping: Aa:Ap=>*:* to Na:Np=>*:*  
• incoming filtering: *:*=>Na:Np to *:*=>Aa:Ap
(IP) restricted cone

• outgoing mapping: Aa:Ap=>*:* to Na:Np=>x:* (remember x)
• incoming filtering: x:*=>Na:Np to x:*=>Aa:Ap
(IP and) port restricted cone

Aa:Ap => Ba:Bp
Ba:Bp => Na:Np
Na:Np => Ba:Bp
Ba:!Bp => Na:Np

• outgoing mapping: Aa:Ap => *:* to Na:Np => x:y (rem x and y)
• incoming filtering: x:y => Na:Np to x:y => Aa:Ap
Symmetric NAT

Aa:Ap => Ba:Bp
Ba:Bp => Aa:Ap
Na:Np => Ba:Bp
Ba:Bp => Na:Np

• outgoing mapping: Aa:Ap => Ba:Bp to Na:Np => Ba:Bp
Why NAT breaks things?
NAT traversal approaches

- Manual configuration
  - static port forwarding at NATs (always open)
- Application layer gateway (ALG)
  - proxy or snoop at NATs
  - application-specific
With NAT cooperation

- Universal Plug ’N Play (UPnP)
  - UPnP-aware NATs and clients
  - security, cascaded NAT, etc.

Open a hole for B to me
Without NAT cooperation

• Simple Traversal of UDP thru NATs (STUN)
  – probe and learn allocated address/port at NATs
  – work with *many* but not all NATs
How about one more NAT?

- **Traversal Using Relay NATs (TURN)**
  - request to allocate address/port at this NAT
  - act as a masquerade relay
Trial and error ...

- Interactive Connectivity Establishment (ICE)

STUN

TURN

• peer-to-peer STUN

• still not bullet-proof!
SIP-assisted NAT traversal

• Why SIP
  – SIP is otherwise NAT-challenged
  – SIP is flexible and extensible
  – SIP may become pervasive

• How NAT traversal with SIP
  – be aware of the existence of NATs
  – determine the type of NATs of the most interest
  – establish sessions btw UAs w/ the help of proxy
UA-Proxy NAT traversal

• Symmetric Response Routing (SRR)
  – UA (x.x.x.x:x)
  – Proxy
    • return received=y.y.y.y;rport=y in SIP attributes
  – UA: if x!\(=\)y, there is NAT(s)!

```
UA: if x!=y, there is NAT(s)!
```

```
UA (x.x.x.x:x)
Proxy
```

```
UA-Proxy NAT traversal
```

```
UDP tunnel
keep alive
```

```
IPsec
```

```
x=>y
```

```
register/auth/invite
```

```
19
```
UA-Proxy STUN: cone

switch addr and port

•cone!
UA-Proxy STUN: symmetric

- symmetric!
- different received: rport

switch addr and port

UA → NAT → Proxy

short timeout
UA-Proxy STUN: restricted

- Switch addr and port
- Short timeout
  - Same received: rport
  - Port restricted!
- Otherwise, (IP) restricted!
### UA-UA: 4x4

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>cone</th>
<th>IP restricted</th>
<th>port restricted</th>
<th>symmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>cone</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>IP restricted</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>port restricted</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>symmetric</td>
<td></td>
<td></td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>

![Diagram showing UA-UA connections](image-url)
UA-UA: cone/restricted-restricted

A UA
Aa:Ap=>Pa:Pp
Na:Np
B at Ma:Mp
A at Na:Np
Ma:Mp
Ba:Bp=>Pa:Pp

Aa:Ap=>Ma:Mp

B at Ma:Mp

Na:Np

Ba:Bp=>Na:Np

Na:Np=>Ba:Bp

timeout

keepalive

media

•ICMP messages?
UA-UA: cone/restricted-symmetric

Ma:M’p=>Aa:Ap  Ma:M’p

keepalive  media
A close look at symmetric NATs

- many symmetric NATs have predictable port allocation
UA-UA: symmetric-symmetric

A [UA] NAT Proxy NAT UA B

probe
predict
using B’s predict
using A’s predict

Na:N’p
Ma:M’p

keepalive
media
How about TCP

- connect()
- ISNa
- SYN
- ACKaSYNb
- ACKb
- ISNb
- listen()

- passive listen()
- active connect()
- sequence number matters!
TCP: more issues

- UDP-encapsulated TCP/IPsec NAT traversal
  - port uniqueness
- Port allocation at UA
  - TCP-based
- Port allocation at NAT
  - UDP-based
- mix and match
  - multi-UA behind the same NAT
SIP-assisted NAT traversal

• SIP becomes versatile
  – such as XML/HTTP for data transfer
• SIP protocol can be extended to support NAT traversal
  – more signaling attributes
• SIP proxy can play an important role in assisting NAT traversal
  – already exists; may become ubiquitous; why not use it for extra purpose?
Thanks!

• Q&A?