AN ANALYSIS OF THE SKYPE PEER-TO-PEER INTERNET TELEPHONY PROTOCOL

By Salman A. Baset and Henning Schulzrinne, Columbia University
Outline

- Skype Overview/Network and NAT Refresher
- Experimental Setup
- Skype Components
- INFOCOMM ‘06 Paper
- Conclusions
Skype Overview

- Developed by Kazaa
- VoIP client with support for (at time of paper):
  - Voice calling
  - Instant messaging
  - Audio conferencing
- Overlay peer-to-peer network with global indexing
- Able to traverse NAT and firewalls
- 256-bit AES Encryption
The Skype Network

- Ordinary Host
  - Skype Client

- Super Node
  - Also a Skype Client
  - Must have a public IP address
  - Determined to have sufficient bandwidth, CPU, memory

- Login Server
The Skype Network (cont’d)
NAT Refresher

- Originally a “quick fix” for limited IPv4 addresses
- Re-mapping of network addresses at the router
Port-restricted NAT

- Assume Host A is behind a port-restricted NAT and Host B is behind a Public IP.
- Host B, which sits on Port P, can only communicate with Host A if Host A has previously sent a packet to Host B on Port P.
- What happens when Host B wants to call Host A?
  - This is a problem for Peer-to-Peer!
Outline

- Skype Overview/Network and NAT Refresher
- Experimental Setup
- Skype Components
- INFOCOMM ‘06 Paper
- Conclusions
Experimental Setup

Borrowed from INFOCOMM '06 Talk
Experimental Setup (cont’d)

Software Tools

- Skype v0.97.0.6 (Windows)
  - Skype was reinstalled for each experiment
  - As of 10/3/09, current Windows version is 4.1
- NCH Tone Generator
  - Generated frequencies to measure the codec range
- Ethereal network protocol analyzer (Wireshark)
  - Captures all traffic passing over a network
- NetPeeker
  - Used to tune bandwidth levels
Experimental Setup (cont’d)
Hardware and Network

- Pentium II 200MHz with 128MB RAM
- Windows 2000
- 10/100 Mb/s ethernet card
- 100 Mb/s network connection
Outline

- Skype Overview/Network and NAT Refresher
- Experimental Setup
- Skype Components
- INFOCOMM ‘06 Paper
- Conclusions
Installation and Startup

- No default ports
  - Random listening port selected at install

- Install
  - GET /ui/0/97/en/installed HTTP/1.1

- Startup
  - GET /ui/0/97/en/getlatestversion?ver=0.97.0.6 HTTP/1.1
Login

- On the first login, Skype client establishes TCP connection with Bootstrap SuperNode
  - Hard-coded into Skype client application
- Logins are routed through a SuperNode
  - If no SuperNodes are reachable, login fails
- Attempts to use Ports 80 and 443 if behind firewall
Login (cont’d)

Start

Send UDP packets to seven bootstrap SNs at port 33033

Response within 6 seconds

TCP connection attempts with seven bootstrap SN IP addresses and port 33033

Connected

Success

Wait for 6 seconds
Host Cache

- Local table of reachable nodes
  - These are actually “SuperNodes”
  - Host cache is populated on the first login
  - Dynamic; SNs are added/dropped as Skype runs
Login (cont’d)
Public IP and NAT

- **SC->BN UDP Connection**
  - SC makes a UDP connection with 80.160.91.12:33033 (Bootstrap node).
  - SC makes a UDP connection with 18B.
  - SC makes a UDP connection with 18B.

- **SC->SN TCP Connection**
  - SC makes a TCP connection with 66.235.180.9. This node becomes a SN.
  - SC makes a TCP connection with 14B.
  - SC makes a TCP connection with 14B.
  - SC makes a TCP connection with 34B.
  - SC makes a TCP connection with 146B.
  - SC makes a TCP connection with 67B.

- **SC->Login Server Auth**
  - SC makes a TCP connection with 80.160.91.11 (Login server).
  - SC makes a TCP connection with 14B.
  - SC makes a TCP connection with 14B.
  - SC makes a TCP connection with 176B (password exchange).
  - SC makes a TCP connection with 246B.
  - SC makes a FIN connection.
  - SC makes a FIN, ACK connection.

- **3-7 seconds**
Login (cont’d)

Mystery ICMP Packets

- Sent during initial login, and not subsequent

```
<table>
<thead>
<tr>
<th>SC</th>
<th>66.235.180.9:33033 (Bootstrap node)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP:SYN</td>
<td>TCP:ACK</td>
</tr>
<tr>
<td>TCP</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP:SYN</td>
<td>TCP:ACK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SC</th>
<th>80.160.91.11 (Login server)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>204.152.229.231</td>
</tr>
<tr>
<td>ICMP</td>
<td>130.244.201.151</td>
</tr>
<tr>
<td>ICMP</td>
<td>202.139.199.243</td>
</tr>
<tr>
<td>ICMP</td>
<td>202.232.43.7</td>
</tr>
</tbody>
</table>
```
Login (cont’d)

Additional UDP Messages

SC sends UDP packets to 4 distinct nodes and receives response over UDP. We believe that these nodes also run Skype.

- Not entirely clear what these are for
  - Remember that all Skype traffic is encrypted – we can’t just inspect the packets
  - Possibly to announce the node’s presence on the Skype network
  - Possibly to determine NAT type (STUN)
STUN Protocol

- Session Traversal Utilities for NAT [RFC 5389]
- Commonly used by networked applications to determine the type of NAT/firewall they are behind
- Requires a STUN server (outside the NAT)
- Determined that there is no centralized STUN server used by Skype
  - So we can infer that Skype clients have STUN client and server functionality
Login (cont’d)
UDP-Restricted Firewall

- **UDP Fails**

- **TCP to Bootstraps**

- **Select SuperNode**

- **UDP Fails Again**

- **Login Server Auth**

- **34 seconds**

Bootstrap nodes 66.235.180.9 and 66.235.181.9 are represented by labels (1) and (2) respectively in subsequent flows.

SC sends UDP packets to 4 distinct nodes. Since it is behind UDP restricted firewall, it cannot receive any responses over UDP.

SC 80.160.91.12:33033 (Bootstrap node)

SC 66.235.181.9:33033 (Bootstrap node) (1)

SC 66.235.180.9:33033 (Bootstrap node) (2)

SC 18B

SC 18B

SC 18B

SC 18B

SC 1B7B

SC 16B

SC 66.235.181.9:33033 (SN)

SC 19B

SC 1205B

SC 407B

SC 80.160.91.11 (Login server)

SC 18B N1, N21, N3, N4

TCP:SYN

TCP:ACK

34 seconds
After Authentication:
Alternate Node Table Construction

SC sends UDP packets to 22 distinct
nodes and receives response from
them over UDP

<table>
<thead>
<tr>
<th>SC</th>
<th>66.235.180.9:33033 (SN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>34B (17 distinct nodes)</td>
</tr>
<tr>
<td>UDP</td>
<td>44B (5 distinct nodes)</td>
</tr>
<tr>
<td>UDP</td>
<td>11B (replies from 22 nodes)</td>
</tr>
</tbody>
</table>

- Conjectured that these are alt nodes
- Confirmed by further communication during call establishment
- Used as replacement SuperNodes
User Search

- Skype guarantees the ability to find any user who has logged on in the past 72 hours
  - Confirmed by experiments
- Decentralized search algorithm
  - Does not involve use of login server
- Intermediate node caching of search results
User Search (cont’d):
Public IP/NAT
User Search (cont’d):
UDP-Restricted Firewall

* SuperNode performs search

83x91
131x91
261x361
357x361
231x361
291x361
327x361
393x349
442x349

Intermediate Node Search Caching

- Local caches cleared on User B client

Diagram:

1. User A searches for "User B" (6-8 secs)
2. User B's local cache is cleared.

Search Results

Diagram:

1. User A searches for "User B" (3-4 secs)
2. User B's local cache is not cleared yet.

Search Results
Call Signaling

- TCP Challenge-Response Mechanism
- If calling a non-buddy, search function is first performed
- At the end of the Call Signaling phase, the Callee’s Skype client will “ring”
Call Signaling (cont’d)

Public IP

Caller press dial

Caller

TCP:SYN
TCP:ACK
TCP
TCP
TCP
TCP
TCP
TCP
TCP
TCP
TCP

Callee

14B
14B
77B
4B
4B
528B
4B
946B
479B

Callee rings
Call Signaling (cont’d)
Caller Behind NAT

- Another online node relays TCP signaling packets
Media Transfer

- Internet Speech Audio Codec (iSAC)
- Frequency range: 50-8000Hz
- Public IPs communicate directly
  - NAT users use a media proxy
- Uses UDP Transport if possible
  - 67 byte UDP voice packets
  - 5 kilobytes/sec
  - UDP-restricting firewall users communicate over TCP
- No Silence Suppression
Why No Silence Suppression?

- Voice packets continue flowing during periods of silence
- For UDP connections, this allows Skype to maintain NAT bindings
- For TCP connections, it is ideal to avoid drops in congestion window
On the use of proxy nodes…

- Enables users behind NAT and Firewall to talk
- Natural solution for conferencing
- However, creates lots of traffic on the proxy
  - Remember, these are regular (mostly unsuspecting) Skype users!
Conferencing

- A: 2GHz P4 w/ 512MB RAM
- B, C: 300MHz P2 w/ 128MB RAM
- “Mixer” elections – A always wins
Additional Findings

- If a Skype call is put on “hold,” a packet is sent every 3 seconds (think lack of silence suppression)
- 2-byte SN Keep-Alive messages every minute
- To maintain reasonable call quality, Skype needs roughly 4 KB/s of available bandwidth
- The same user can log in from multiple machines simultaneously
- Buddy lists stored locally
- Cannot select your own SuperNode by manually populating the Host Cache with a Skype Client’s IP
Outline

- Skype Overview/Network and NAT Refresher
- Experimental Setup
- Skype Components
- INFOCOMM ‘06 Paper
- Conclusions
INFOCOMM ’06

Overview

- Skype version 1.4 used
- Re-performed experiments
- Comparisons with Yahoo, MSN, GTalk
- Closer look at SuperNodes
INFOCOMM ‘06 (cont’d)
Skype vs Yahoo, MSN, Gtalk (Setup)

- Measurement – Mouth-to-ear Latency

Borrowed from INFOCOMM ‘06 Talk
## INFOCOMM ‘06 (cont’d)
### Skype vs Yahoo, MSN, GTalk (Results)

<table>
<thead>
<tr>
<th>Application version</th>
<th>Memory usage before call (caller, callee)</th>
<th>Memory usage after call (caller, callee)</th>
<th>Process priority before call</th>
<th>Process priority during call</th>
<th>Mouth-to-ear latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skype 1.4.0.84</td>
<td>19 MB, 19 MB</td>
<td>21 MB, 27 MB</td>
<td>Normal</td>
<td>High</td>
<td>96ms</td>
</tr>
<tr>
<td>MSN 7.5</td>
<td>25 MB, 22 MB</td>
<td>34 MB, 31 MB</td>
<td>Normal</td>
<td>Normal</td>
<td>184ms</td>
</tr>
<tr>
<td>Yahoo 7.0 beta</td>
<td>38 MB, 34 MB</td>
<td>43 MB, 42 MB</td>
<td>Normal</td>
<td>Normal</td>
<td>152ms</td>
</tr>
<tr>
<td>GTalk 1.0.0.80</td>
<td>9 MB, 9 MB</td>
<td>13 MB, 13 MB</td>
<td>Normal</td>
<td>Normal</td>
<td>109ms</td>
</tr>
</tbody>
</table>
Mystery ICMP Packets Revisited

- 204.152.* (USA)
- 130.244.* (Sweden)
- 202.139.* (Australia)
- 202.232.* (Japan)

The purpose of these packets is still unclear!
INFOCOMM ‘06 (cont’d)

Super Nodes Revisited

- Automated 8,153 Skype logins over 4 days to analyze SuperNode selection
- Found that the top 20 SNs received 43.8% of total connections

<table>
<thead>
<tr>
<th></th>
<th>Unique SNs per day</th>
<th>Cumulative unique SNs</th>
<th>Common SNs between previous and current day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day1</td>
<td>224</td>
<td>224</td>
<td></td>
</tr>
<tr>
<td>Day2</td>
<td>371</td>
<td>553</td>
<td>42</td>
</tr>
<tr>
<td>Day3</td>
<td>202</td>
<td>699</td>
<td>98</td>
</tr>
<tr>
<td>Day4</td>
<td>246</td>
<td>898</td>
<td>103</td>
</tr>
</tbody>
</table>
INFOCOMM ‘06 (cont’d)
Super Node Map

35% of SuperNodes are from .edu!
INFOCOMM ‘06 (cont’d)
Additional Findings

- Host Cache moved from registry to XML file
- Keep-alive messages are half as frequent
- Buddy Lists now stored on login server
- Voice packets increased to 70-100 bytes
Outline

- Skype Overview/Network and NAT Refresher
- Experimental Setup
- Skype Components
- INFOCOMM ‘06 Paper
- Conclusions
Conclusions

- Search not entirely clear
- Login server is centralized (but nothing else)
- Best mouth-to-ear latency
- ‘Selfish’ application
Further Research

- INFOCOMM paper has 467 citations [Google Scholar]

- PEDS Research Group on 10/5/09: Rapid Identification of Skype Traffic Flows
  - Able to identify Skype traffic by observing 5 sec flow
  - Looks at packet lengths and inter-arrival times
  - 98% precise
  - But – codec-dependent!
Too popular?

- Throttled on WPI campus internet

Re: Skype

Phillip G Deneault to Andrew, netops

Yes, Skype is throttled. It will be a few weeks before we can review usage patterns to adjust for higher levels of service.

Thanks,
Phil

On Sun, 30 Aug 2009, Keating, Andrew wrote:

Hello,

Is Skype traffic being throttled? Myself and others using the on-campus network are seeing very poor Skype quality of service.

Thanks,
Andrew Keating
A Quick Test

- Caller and Callee behind public IP addresses
- Caller on wired WPI campus connection (throttled)
- Callee on unrestricted home network connection

- Video chat UDP:
  - 15-25% packet loss
  - 8.448 kilobytes per second avg.
  - Result – Jitter, “robotic” voice, low QoS

- Video chat TCP
  - 0% packet loss
  - 36.7 kilobytes per second avg.
Questions/Discussion

- Ideas on the mysterious ICMP packets?
- Ideas on how the search algorithm works?
- Why is WPI throttling Skype?
- Feelings about assigning of unsuspecting supernodes?