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

By Salman A. Baset and Henning Schulzrinne, Columbia University

Presentation by Andrew Keating for CS577 Fall 2009

Outline

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

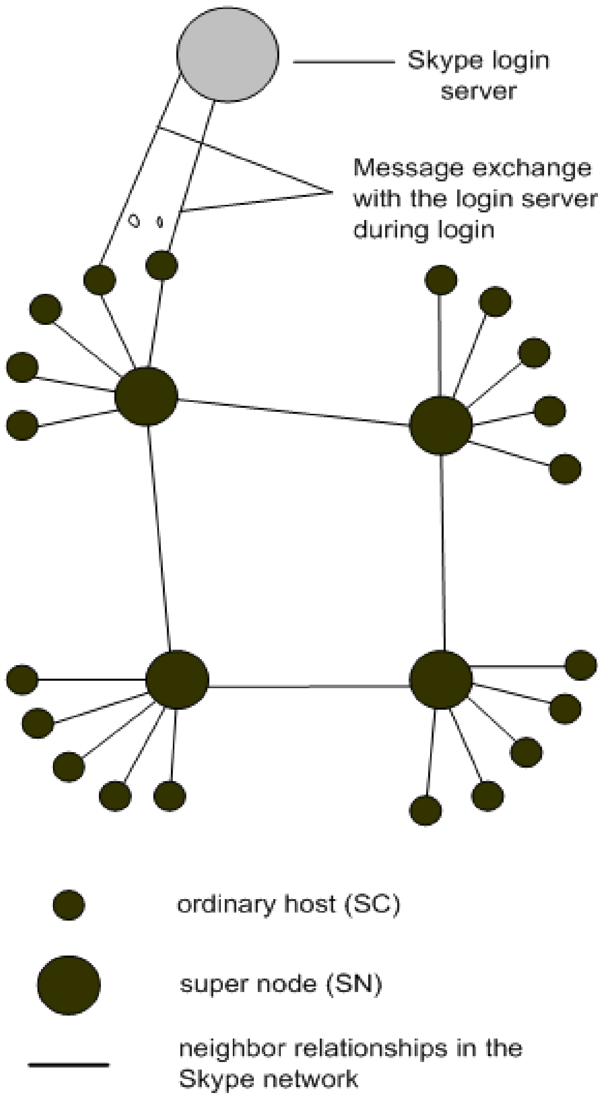
Skype Overview

- 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

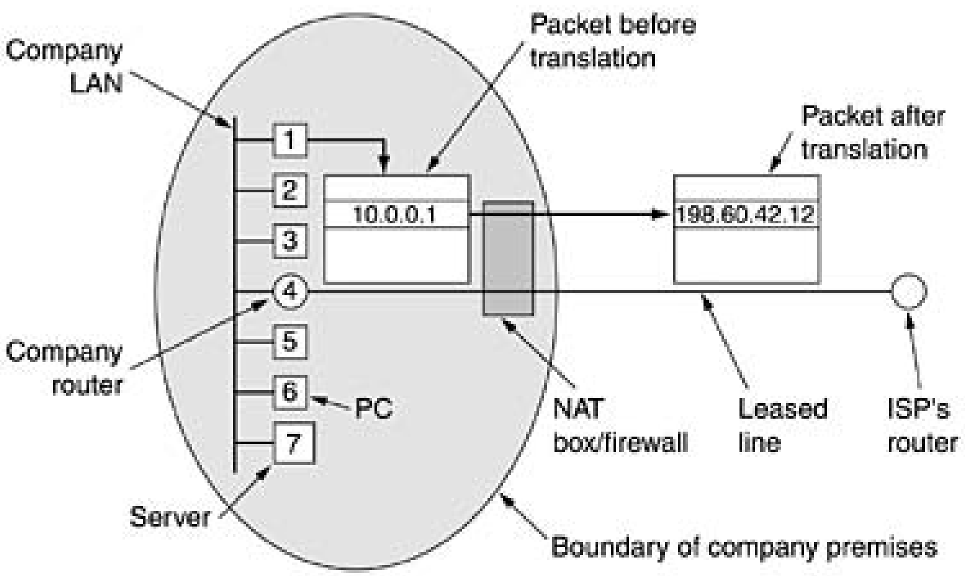
- Ordinary Host
  - Skype Client
- Super Node
  - Also a Skype Client
  - Must have a public IP address
  - Determined to have sufficient bandwidth, CPU, memory
- Login Server

The Skype Network (cont'd)



NAT Refresher

- Originally a "quick fix" for limited IPv4 addresses
- Re-mapping of network addresses at the router



Tanenbaum 4th

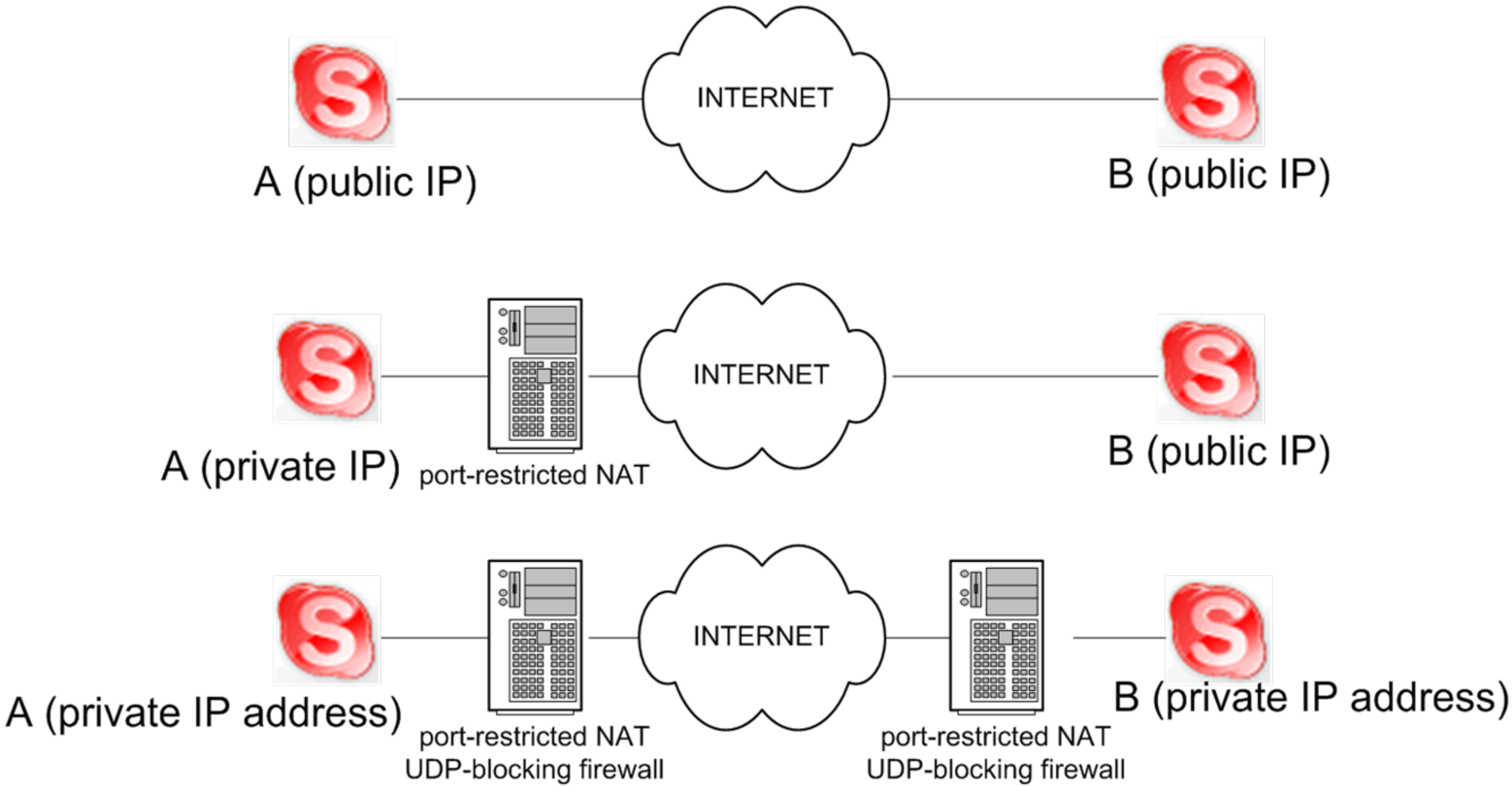
NAT Refresher (cont'd)
Port-restricted NAT

- 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

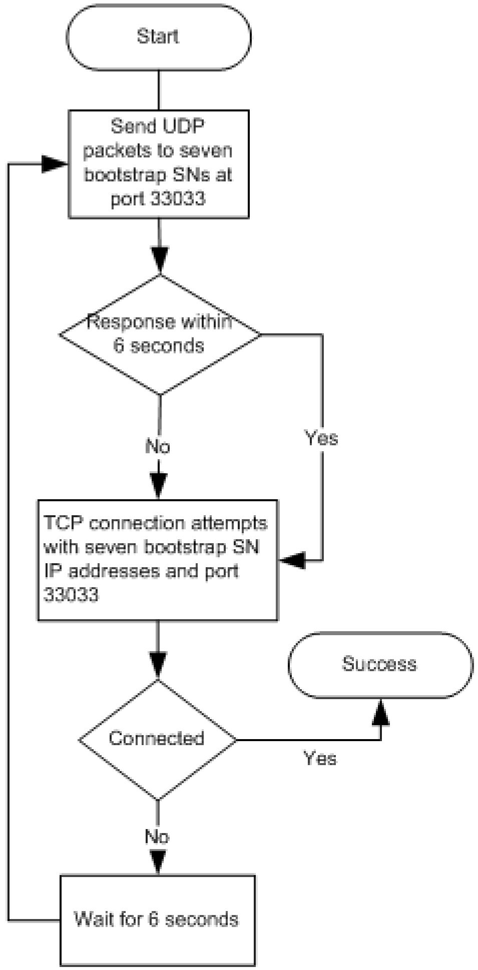
Installation and Startup

- 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

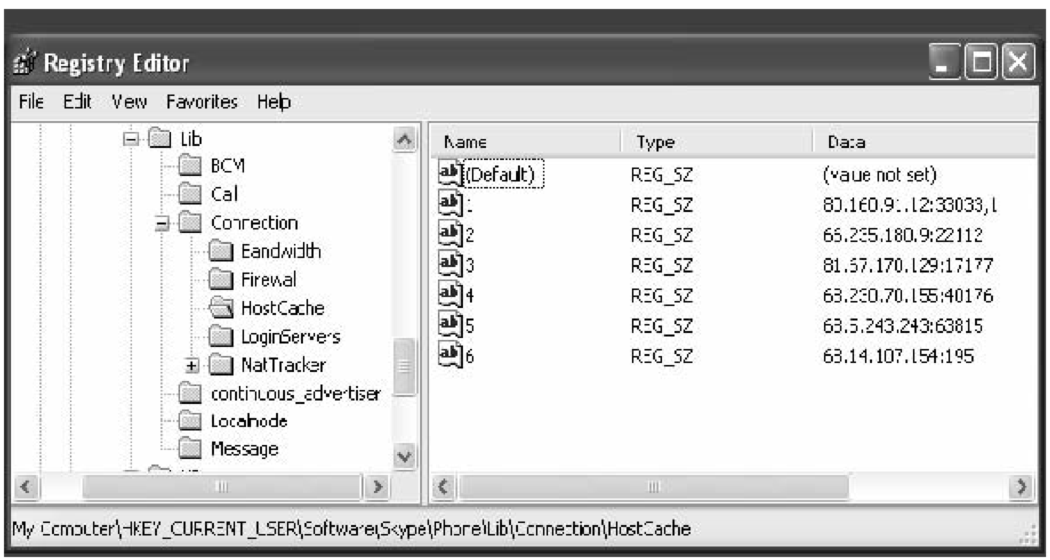
- On the first login, Skype client establishes TCP connection with Bootstrap SuperNode
  - Hard-coded into Skype client application
- Logins are routed through a SuperNode
  - If no SuperNodes are reachable, login fails
- Attempts to use Ports 80 and 443 if behind firewall

Login (cont'd)


Start
Send UDP packets to seven bootstrap SNs at port 33033
Response within 6 seconds
Yes
No
TCP connection attempts with seven bootstrap SN IP addresses and port 33033
Success
Connected
Yes
No
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

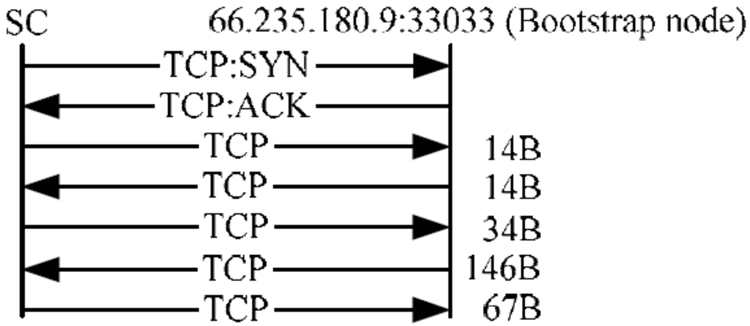


Login (cont'd)
Public IP and NAT

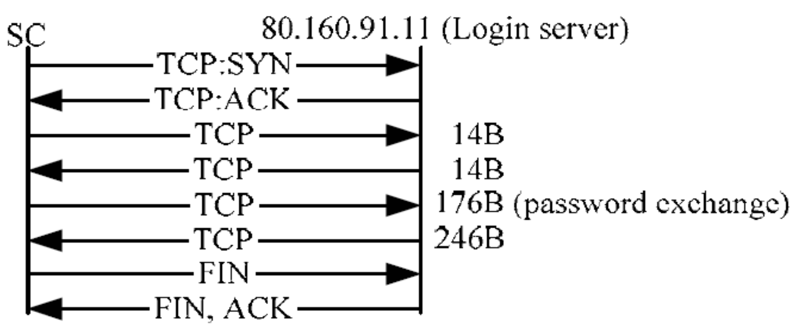
- SC->BN UDP Connection

- SC->SN TCP Connection

SC makes a TCP connection with 66.235.180.9. This node becomes a SN.



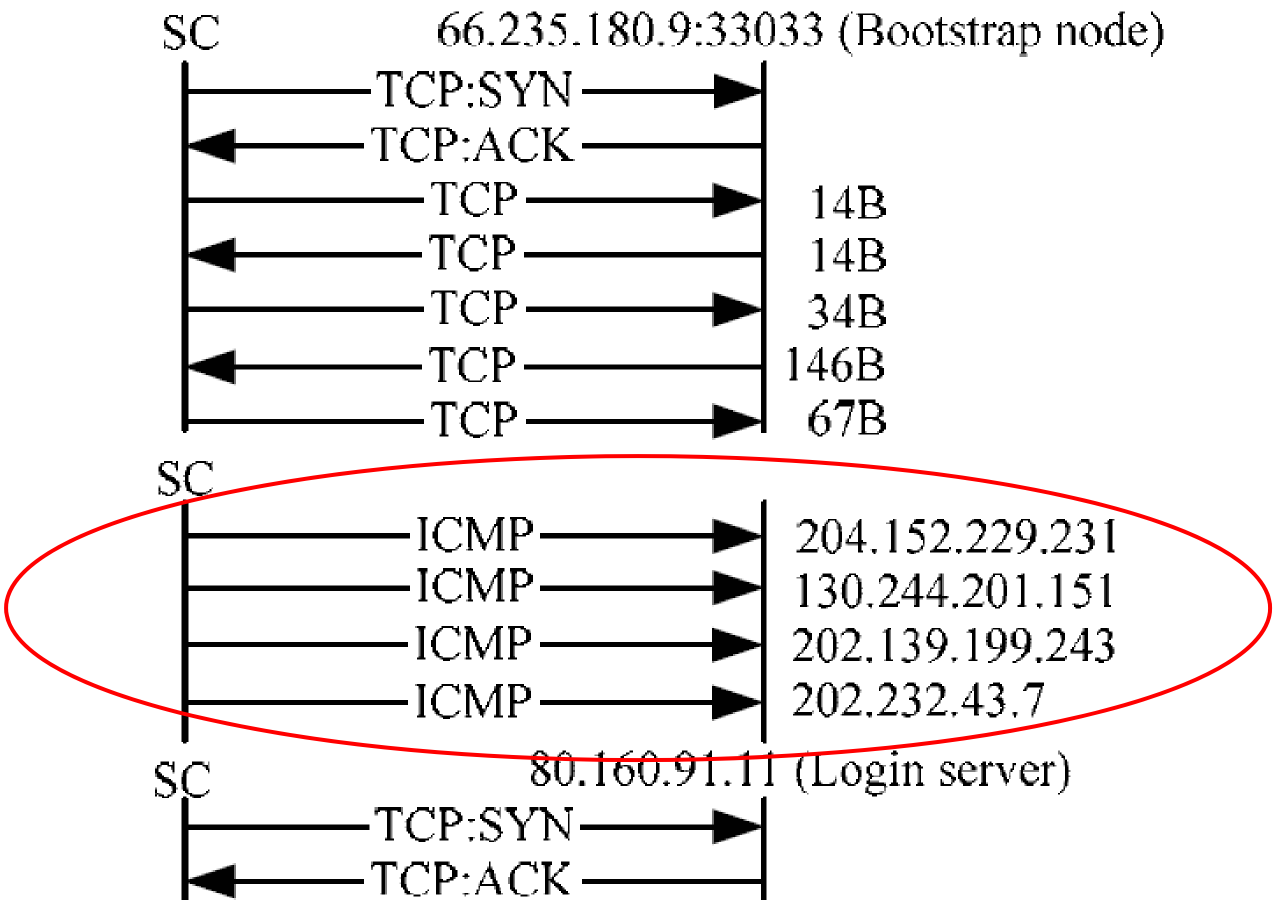
- SC->Login Server Auth



- 3-7 seconds

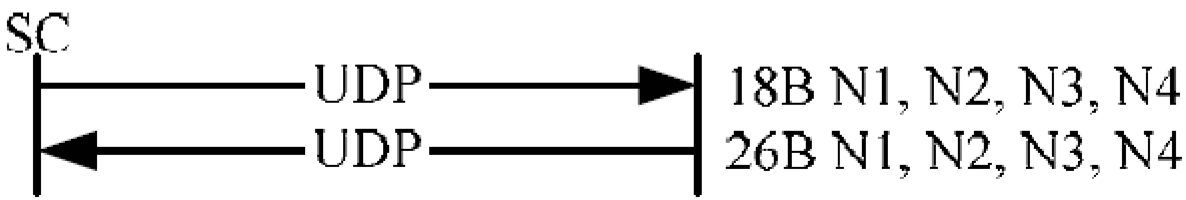
Login (cont'd)
Mystery ICMP Packets

- Sent during initial login, and not subsequent



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.



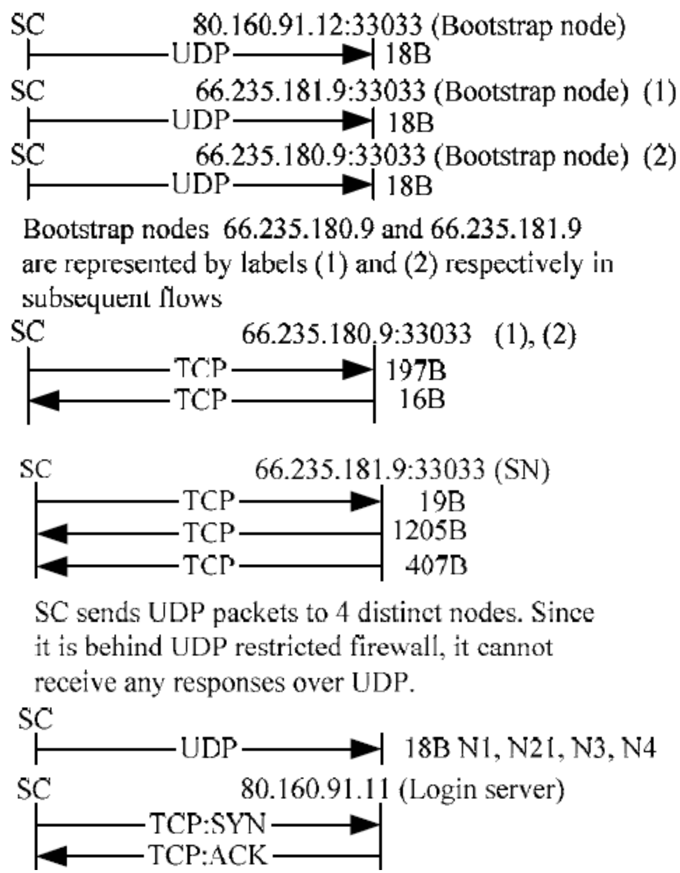
- Not entirely clear what these are for
  - Remember that all Skype traffic is encrypted – we can't just inspect the packets
  - Possibly to announce the node's presence on the Skype network
  - Possibly to determine NAT type (STUN)

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

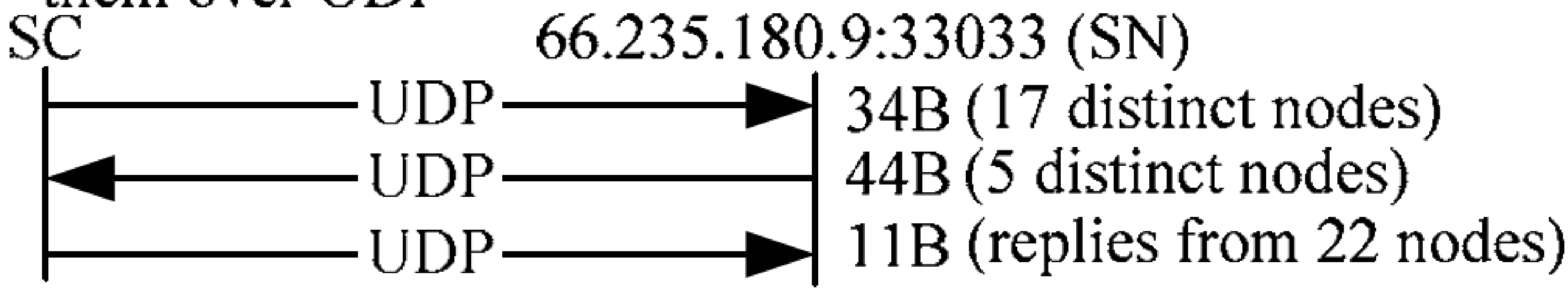
Login (cont'd) UDP-Restricted Firewall

- UDP Fails
- TCP to Bootstraps
- Select SuperNode
- UDP Fails Again
- Login Server Auth
- 34 seconds



After Authentication: Alternate Node Table Construction

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

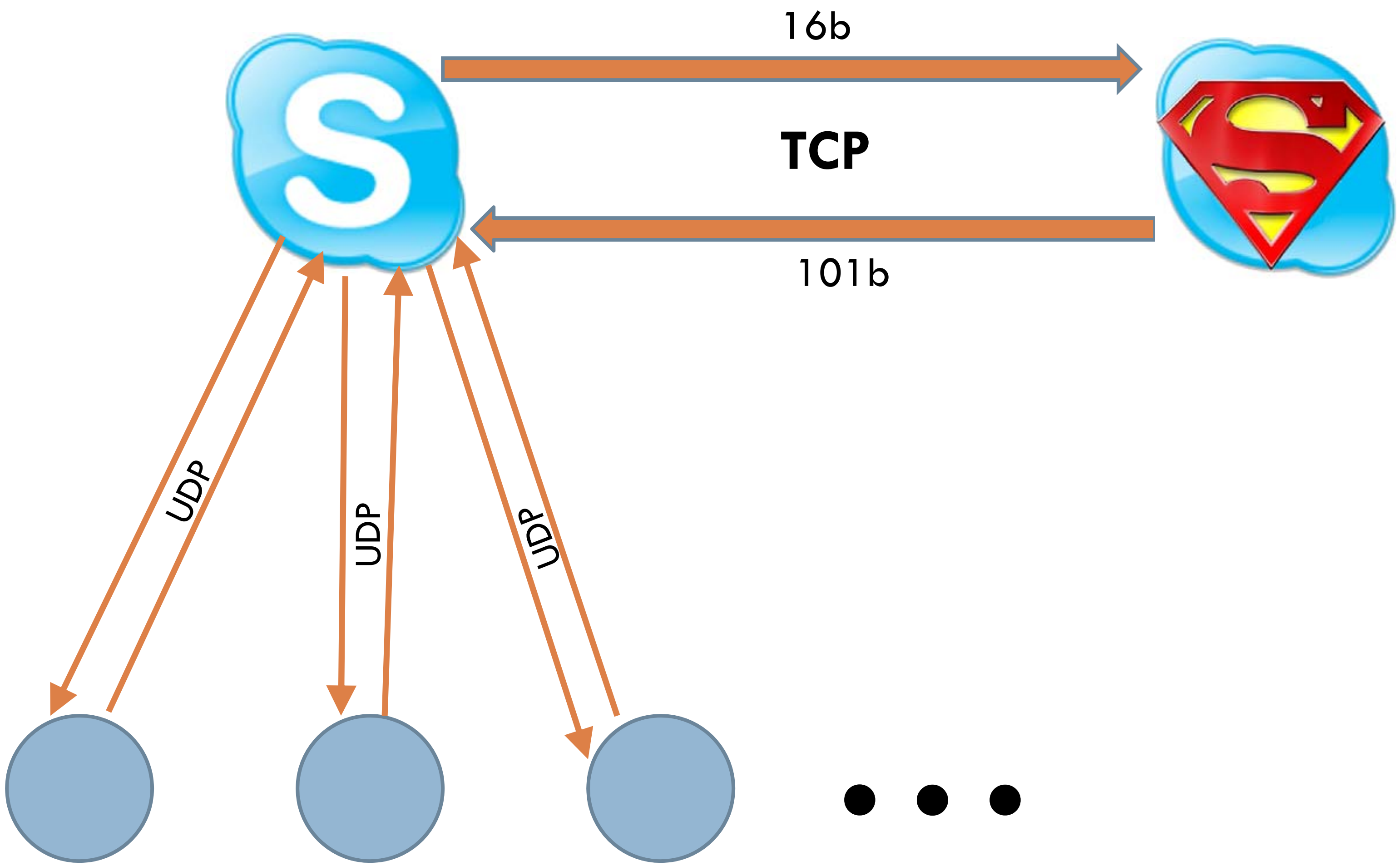


- Conjectured that these are alt nodes
- Confirmed by further communication during call establishment
- Used as replacement SuperNodes

User Search

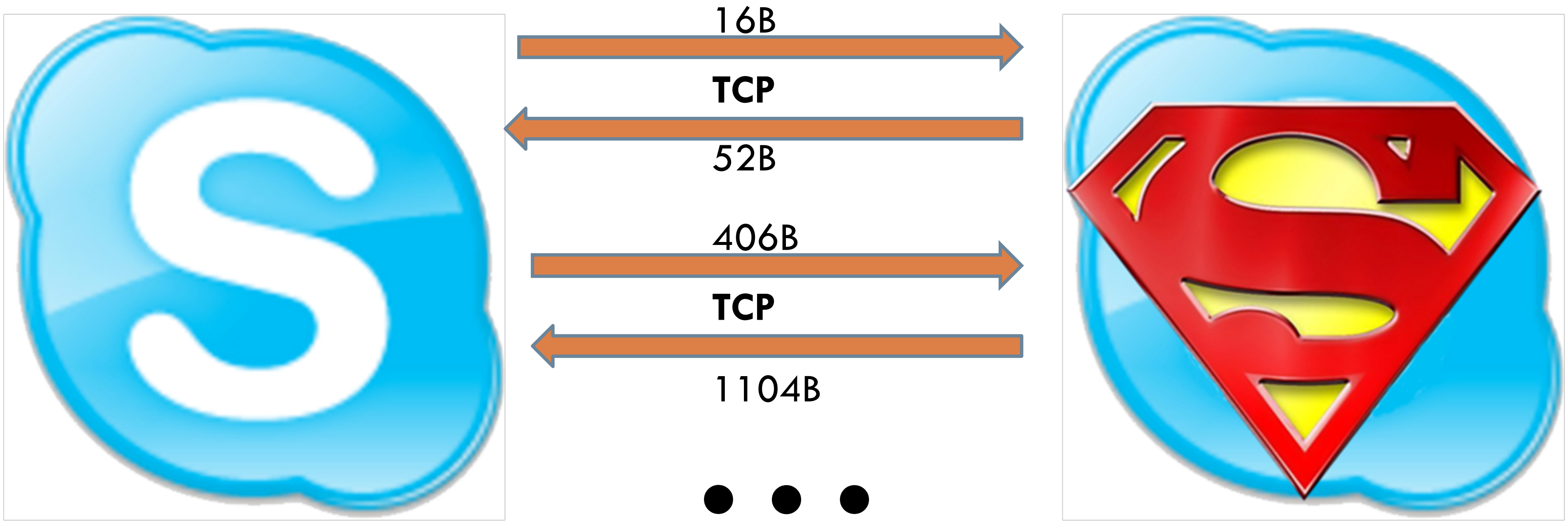
- Skype guarantees the ability to find any user who has logged on in the past 72 hours
  - Confirmed by experiments
- Decentralized search algorithm
  - Does not involve use of login server
- Intermediate node caching of search results

User Search (cont'd): Public IP/NAT



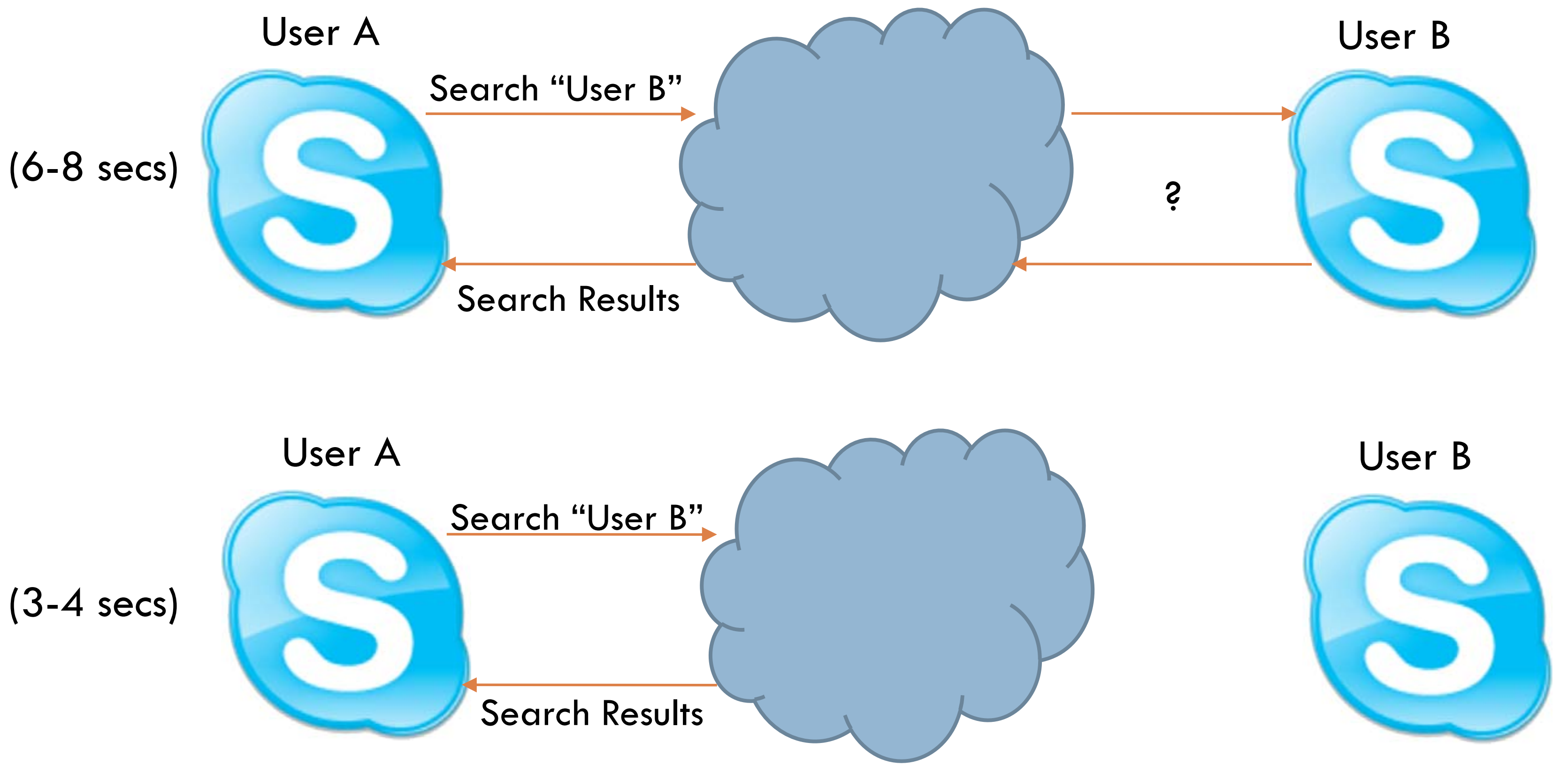
User Search (cont'd): UDP-Restricted Firewall

- SuperNode performs search



Intermediate Node Search Caching

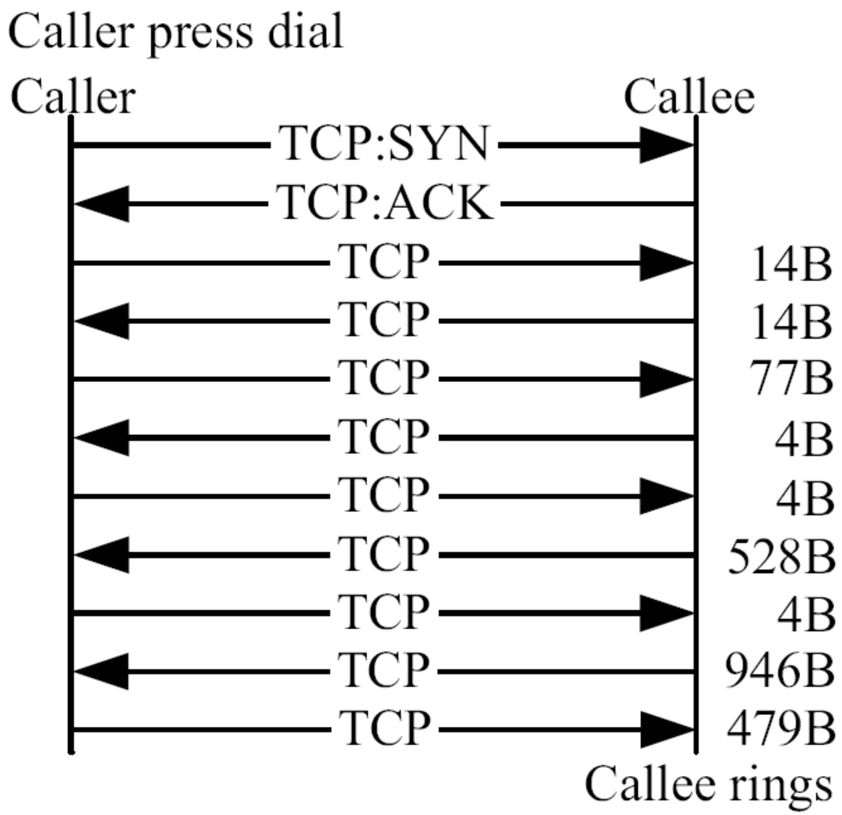
- Local caches cleared on User B client



Call Signaling

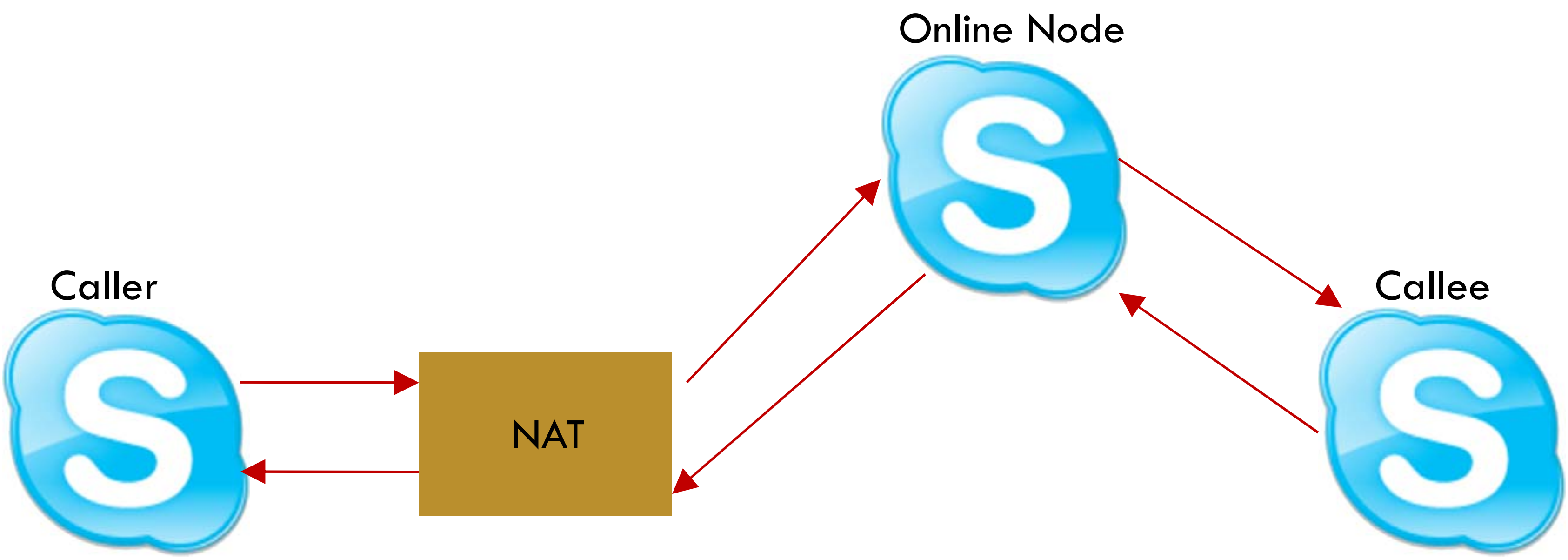
- TCP Challenge-Response Mechanism
- If calling a non-buddy, search function is first performed
- At the end of the Call Signaling phase, the Callee's Skype client will "ring"

Call Signaling (cont'd) Public IP



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?

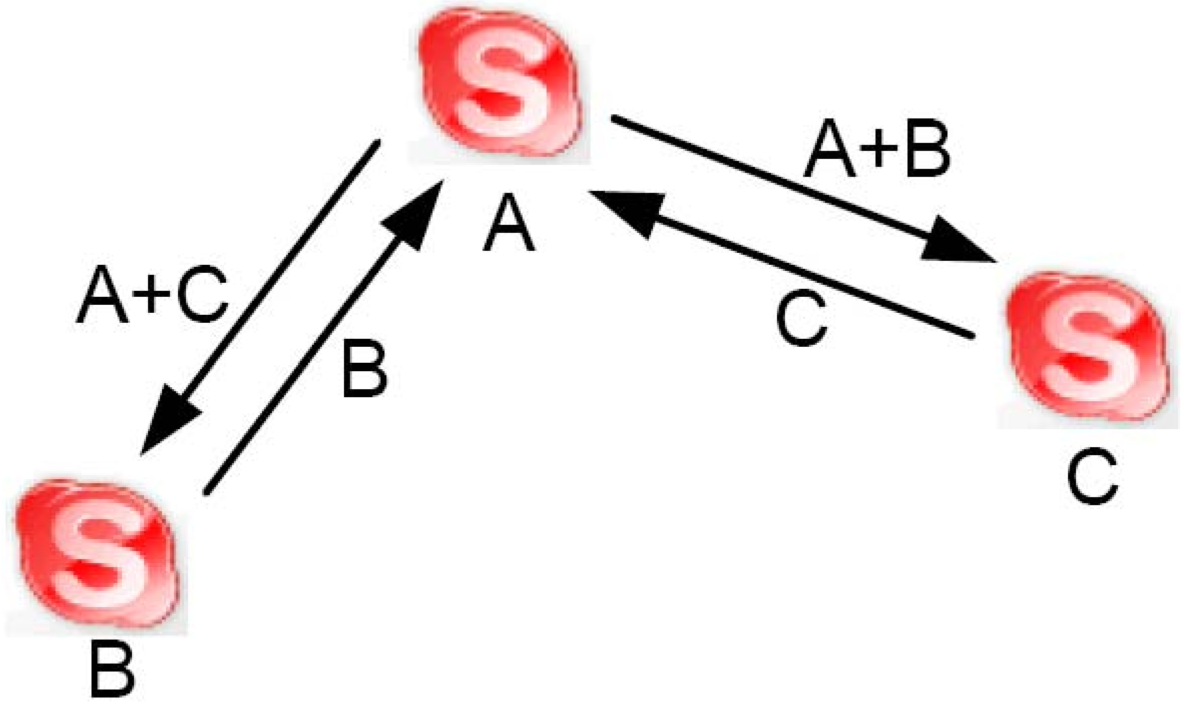
- Voice packets continue flowing during periods of silence
- For UDP connections, this allows Skype to maintain NAT bindings
- For TCP connections, it is ideal to avoid drops in congestion window

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

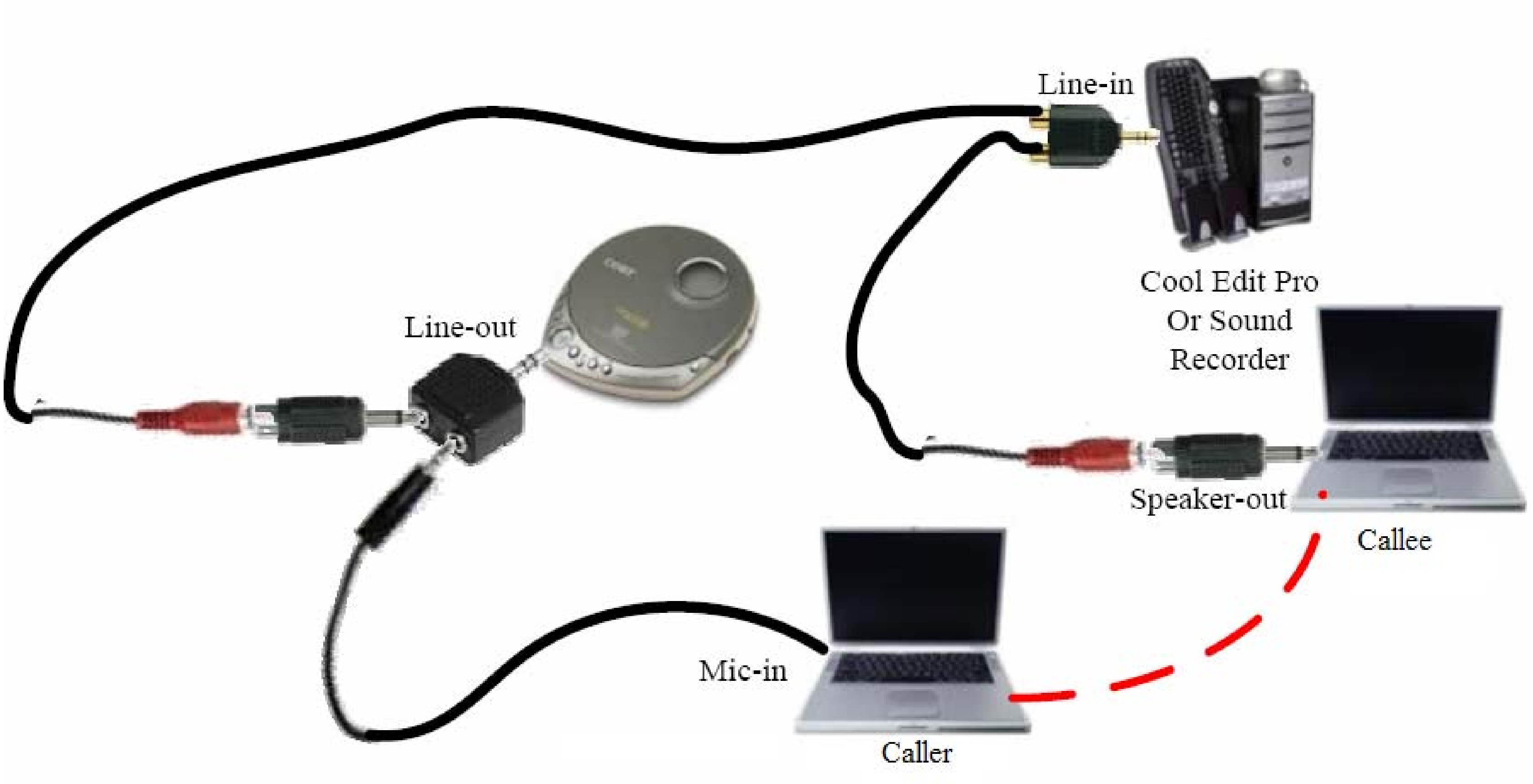
- Skype Overview/Network and NAT Refresher
- Experimental Setup
- Skype Components
- INFOCOMM '06 Paper
- Conclusions

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)

	Application version	Memory usage before call (caller, callee)	Memory usage after call (caller, callee)	Process priority before call	Process priority during call	Mouth-to-ear latency
Skype	1.4.0.84	19 MB, 19 MB	21 MB, 27 MB	Normal	High	**96ms**
MSN	7.5	25 MB, 22 MB	34 MB, 31 MB	Normal	Normal	**184ms**
Yahoo	7.0 beta	38 MB, 34 MB	43 MB, 42 MB	Normal	Normal	**152ms**
GTalk	1.0.0.80	9 MB, 9 MB	13 MB, 13 MB	Normal	Normal	**109ms**

INFOCOMM '06 (cont'd) Mystery ICMP Packets Revisited

- 204.152.* (USA)
- 130.244.* (Sweden)
- 202.139.* (Australia)
- 202.232.* (Japan)
- The purpose of these packets is still unclear!

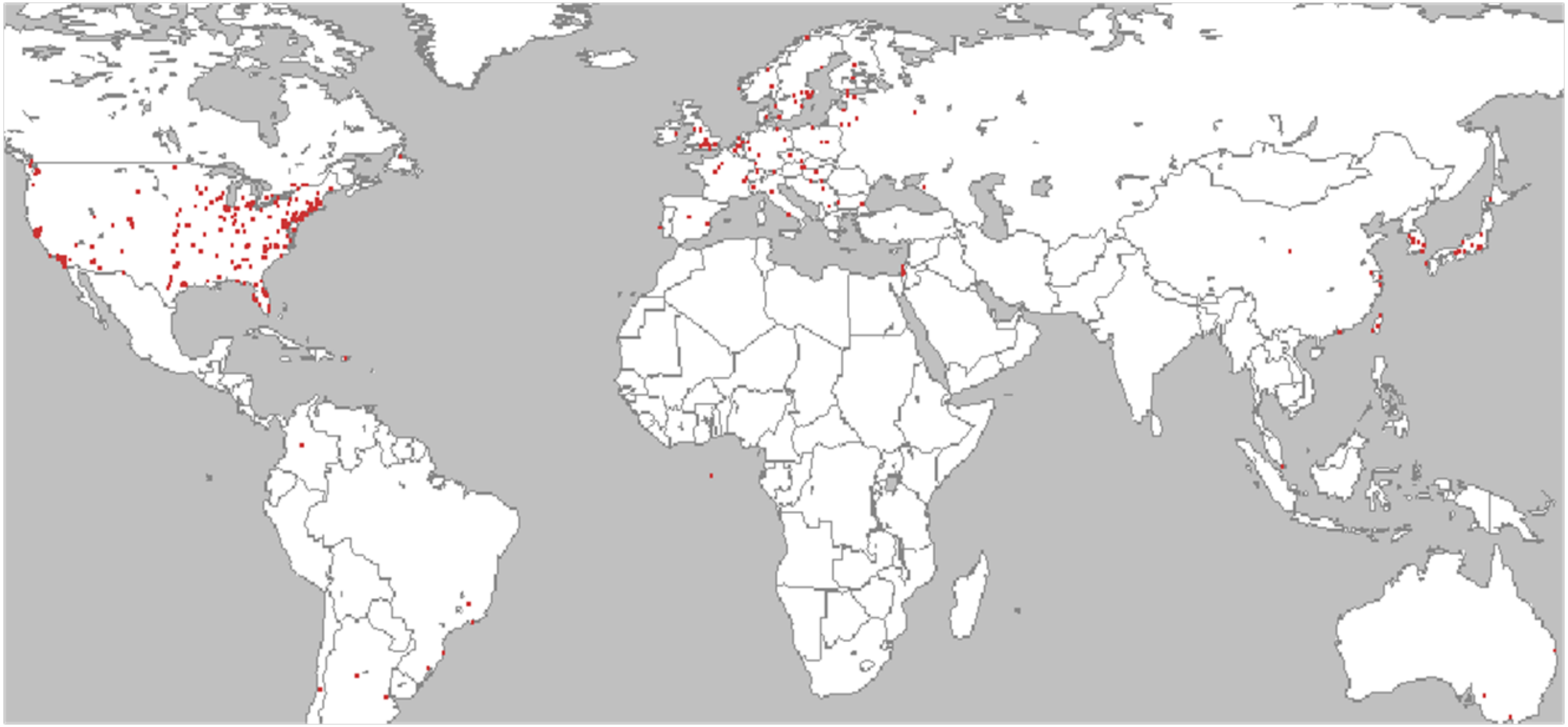
INFOCOMM '06 (cont'd)
Super Nodes Revisited

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

	Unique SNs per day	Cumulative unique SNs	Common SNs between previous and current day
Day1	224	224	
Day2	371	553	42
Day3	202	699	98
Day4	246	898	103

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 Inbox | X

Phillip G Deneault to Andrew, netops

Yes, Skype is throttled. It will be a few weeks before we can review usae 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?