

An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol

NEWS@NTU

Embedded Systems and Wireless Networking Labs

Ai-Chun Pang
**Graduate Institute of Networking and
Multimedia**
Dept. of Comp. Sci. and Info. Engr.
National Taiwan University



NEWS@NTU

References

- **S. A. Baset and H. Schulzrinne, “An Analysis of the Skype Peer-to-Peer Internet Telephone Protocol, ” Dept. of CS, Columbia University, Sep. 15, 2004.**
- **Stephanos Androutsellis-Theotokis and Diomidis Spinellis, “A Survey of Peer-to-Peer Content Distribution Technologies,” Athens University of Economics and Business, ACM Computing Surveys, December 2004.**
- **Skype conferencing white paper by PowerModeling**
 - <http://www.powermodeling.com/files/whitepapers/Conference%20Test%20feb%2009.pdf>
- **Skype Explained**
 - http://www.skype.com/skype_p2pexplained.html



What's Overlay Network & What's P2P ?

What is P2P ?

- **Distributed systems**
- **Direct sharing of computer resources**
- **Without requiring the intermediation or support of a global centralized server or authority.**

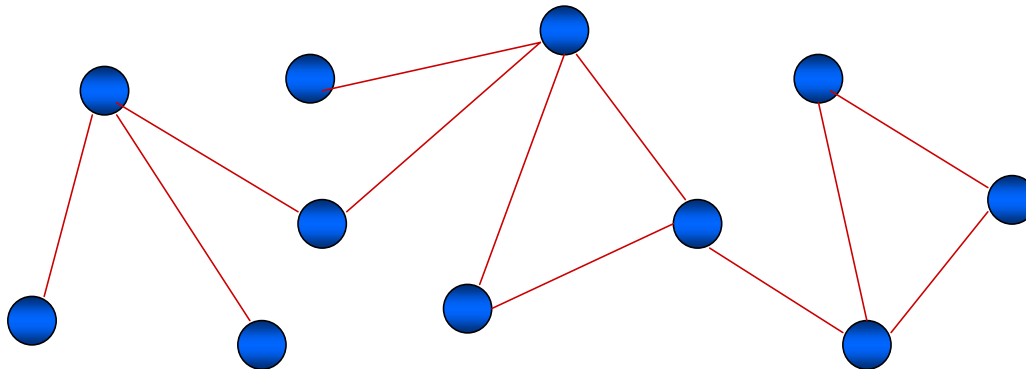
What is Overlay Network ?

- **The operation of any peer-to-peer system relies on a network of peer computers (nodes), and connections (edges) between them.**
- **This network is formed on top of –and independently from–the underlying physical computer (typically IP) network and is thus referred to as an “overlay” network.**

Overlay Network Architecture (1/3)

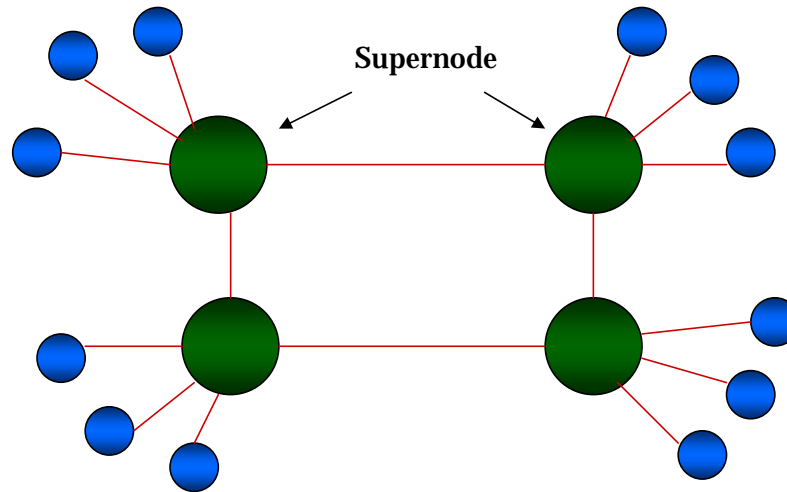
- **Purely Decentralized Architectures**

- All nodes in the network perform exactly the same tasks, acting both as servers and clients, and there is no central coordination of their activities.



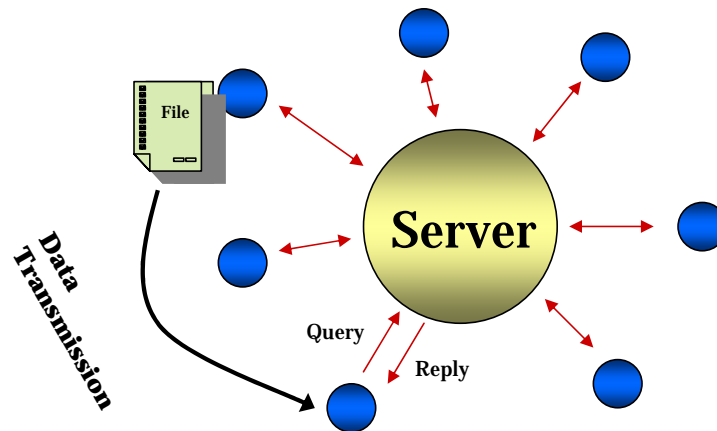
Overlay Network Architecture (2/3)

- **Partially Centralized Architectures**



Overlay Network Architecture (3/3)

- **Hybrid Decentralized Architectures**



Classification of P2P Applications

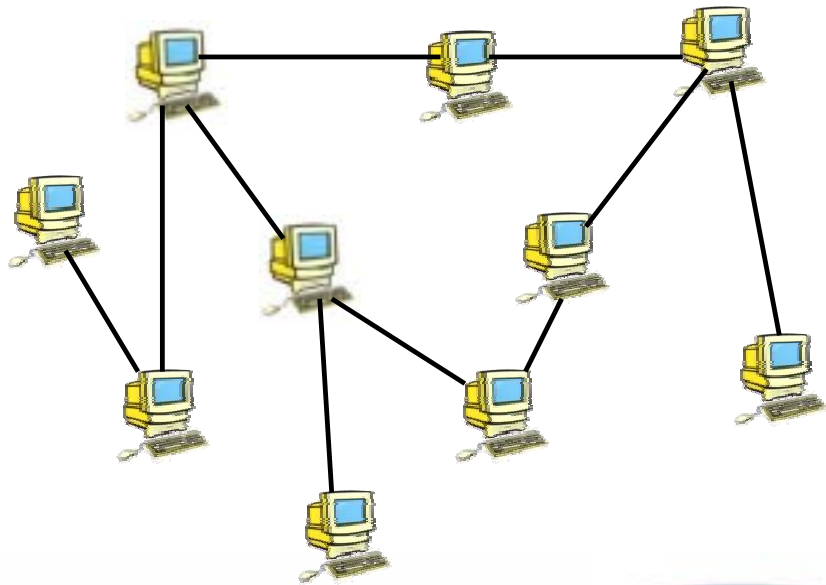
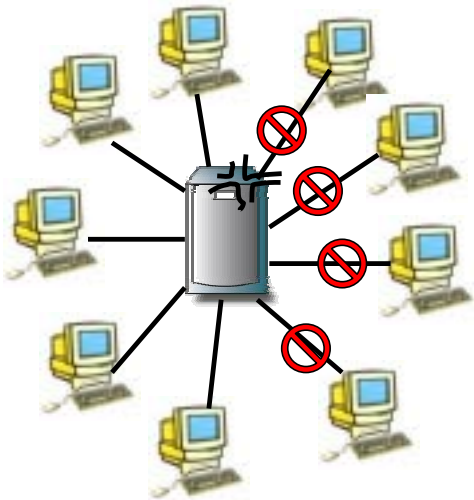
- **Communication and Collaboration**
- **Distributed Computation**
- **Database Systems**
- **Content Distribution**
 - **Peer-to-Peer File Exchange Systems**
 - *Napster* : Hybrid decentralized.
 - *KaZaA* : Partially centralized.
 - *Gnutella* : Purely decentralized.



Advantages of P2P (1/3)

- **Scalability**

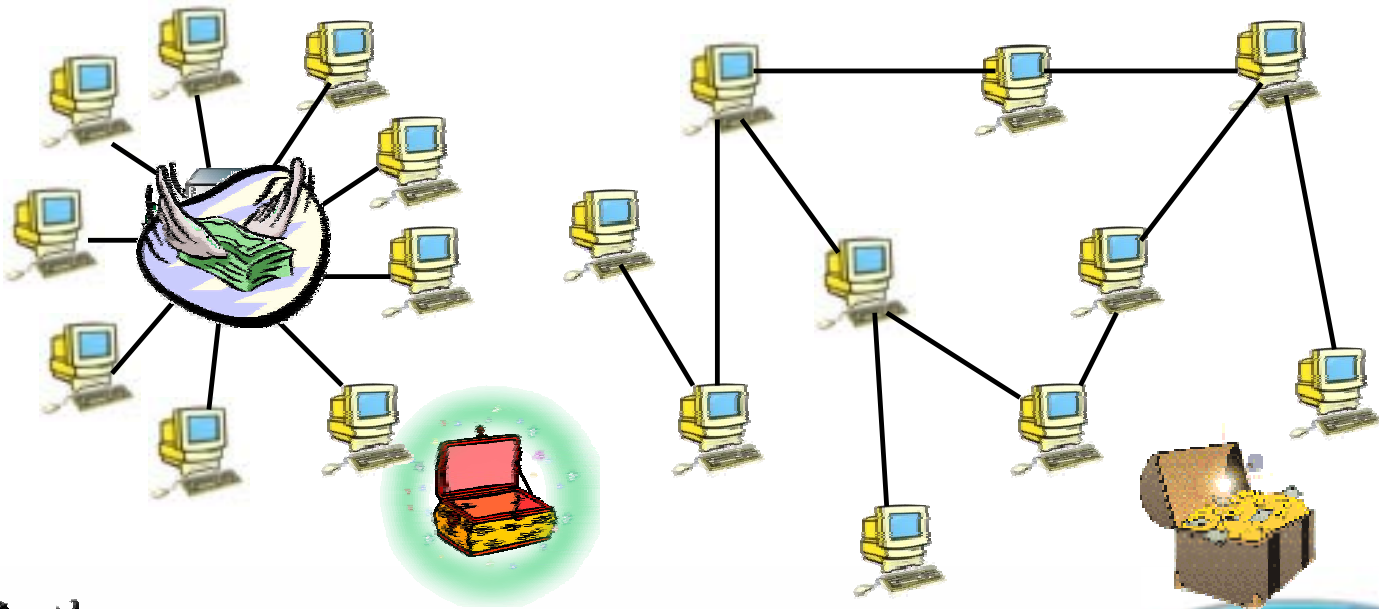
- A dramatic increase in the number of nodes or documents will have minimal effect on performance and availability.



Advantages of P2P (2/3)

- **Low Cost**

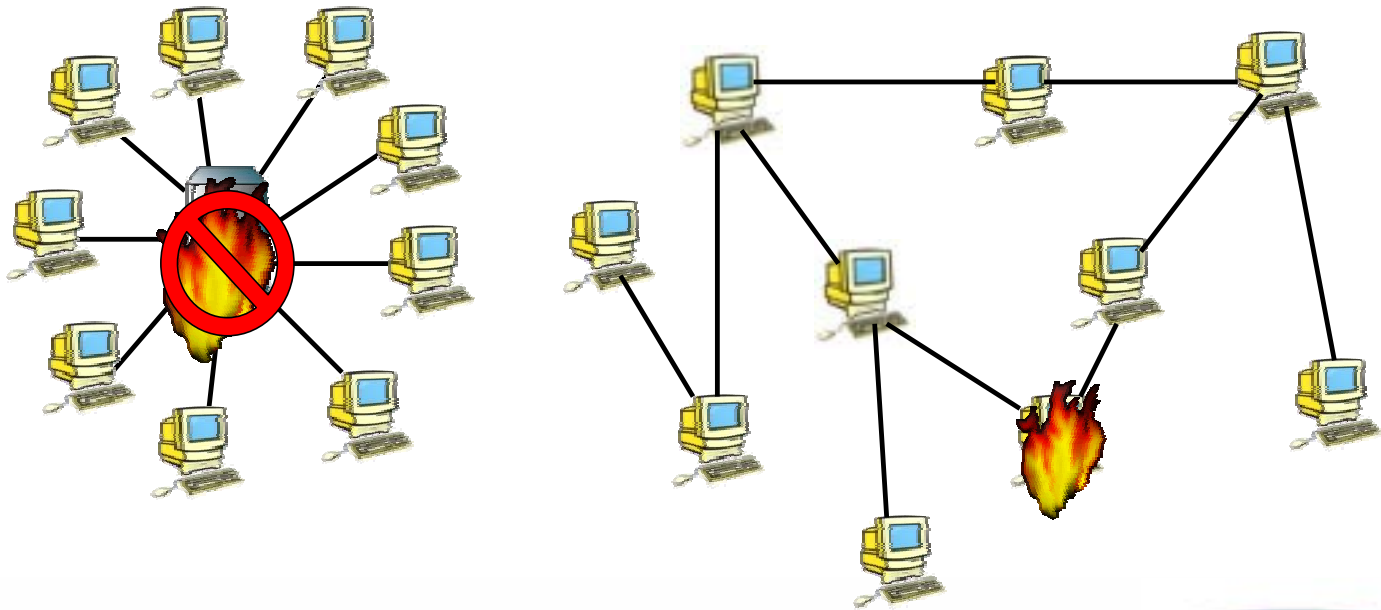
- There is no need to buy more special machines to be servers. Every computer can be a server and a client at the same time.



Advantages of P2P (3/3)

- **Robustness and Reliability**

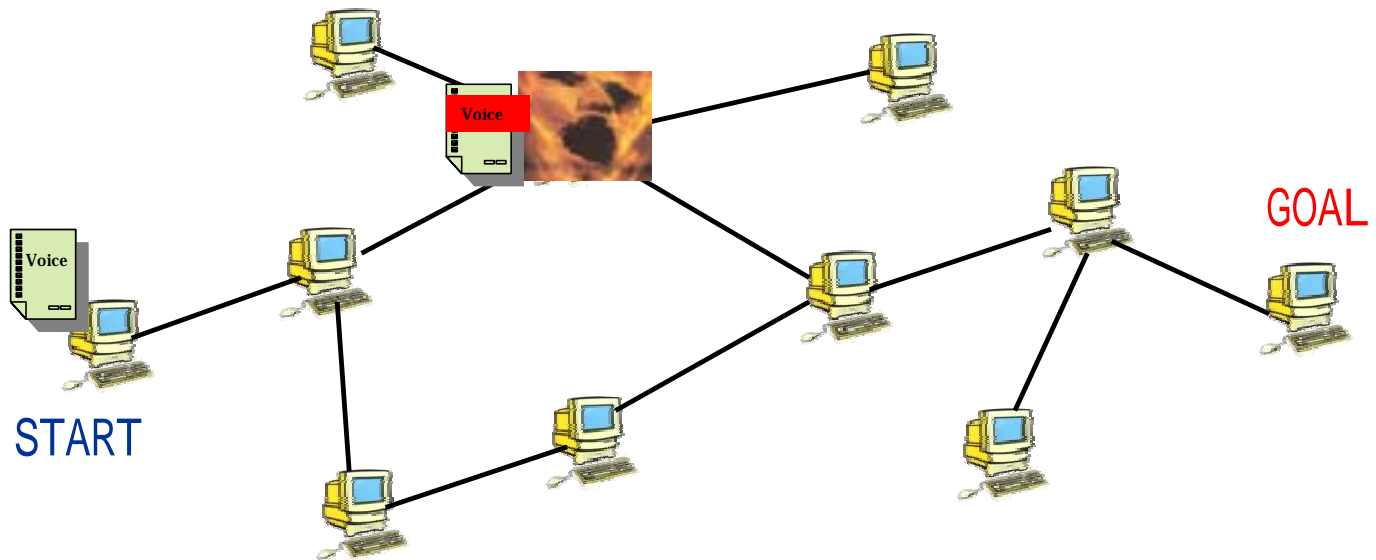
- It could work without centralized server.
- Increased Network Connectivity



Issues of P2P (1/2)

- **Security**

- Integrity and authenticity.
- Privacy and confidentiality.



Issues of P2P (2/2)

- **Performance**

- The time required for performing the operations allowed by the system, typically routing, searching, and retrieval of documents.

- **Fairness**

- Ensuring that users offer and consume resources in a fair and balanced manner.
- Resource Management Capabilities

An Example of Voice over Overlay Network

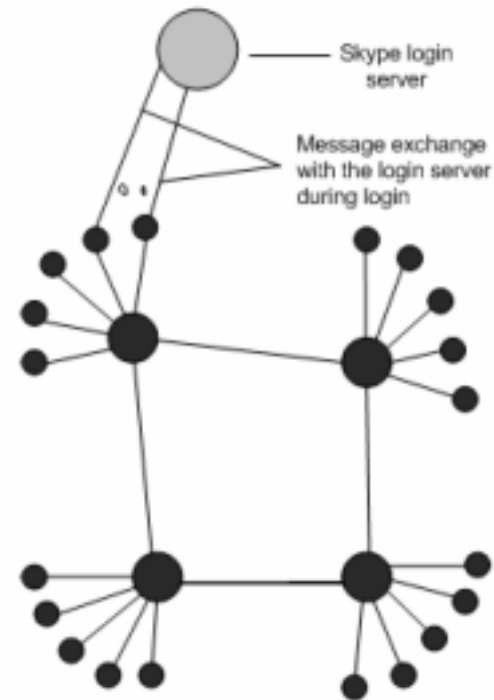
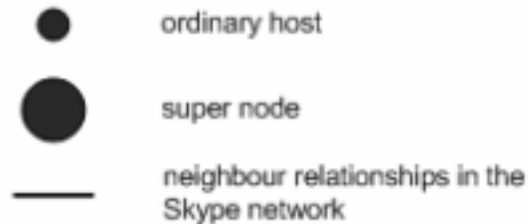
Introduction

- **Skype is a peer-to-peer VoIP client developed by KaZaa in 2003**
- **Skype claims that**
 - It can work almost seamlessly across NATs and firewalls
 - It has better voice quality than the MSN and Yahoo IM applications
- **The key Skype functions include**
 - Login
 - NAT and firewall traversal
 - Call establishment and teardown
 - Media transfer
 - Codecs
 - Conferencing



Skype Network

- Any Skype Client (SC) with a **public IP address** having **sufficient CPU, memory,** and **network bandwidth** is a candidate to become a **super node (SN)**



Key Components of Skype Software [1/2]

- **Ports**

- SC opens a TCP and an UDP listening port
- SC also opens port 80 (HTTP) and port 443 (HTTPS)
- There is no default TCP or UDP listening port

- **Host Cache (HC)**

- The HC is a list of super node IP:Port pairs
- A SC stores HC in the Windows registry at
HKEY_CURRENT_USER / SOFTWARE / SKYPE / PHONE / LIB /
CONNECTION / HOSTCACHE
- HC contains a maximum of 200 entries

- **Codecs**

- The white paper observes that Skype uses **iLBC**, **iSAC**, or a third unknown codec
- Skype codecs allow frequency between 50-8000 Hz to pass through



Key Components of Skype Software [2/2]

- **Buddy List**

- Skype stores its buddy information in the Windows registry
 - Digitally signed and encrypted
- The buddy list is local to one machine and is not stored on a central server

- **Encryption**

- Skype uses AES (Advanced Encryption Standard)
 - 256-bit key (1.1×10^{77} possible keys)
- Skype uses 1536 to 2048 bit RSA to negotiate symmetric AES keys



Experimental Setup

- **Version 0.97.0.6**
 - Latest version 1.0.0.106
- **Under three different network setups**
 - 1) Both Skype users were on machines with public IP address
 - 2) One Skype user was behind port-restricted NAT
 - 3) Both Skype users were behind port-restricted NAT and UDP-restricted firewall
- **Ethereal was used to monitor network traffic**
- **NetPeeker was used to tune the bandwidth**

Skype Functions

- **Startup**

- When SC was run for the first time after installation
 - sent a HTTP 1.1 GET request (contains the keyword “installed”) to the Skype server
- During subsequent startups
 - a SC only sent a HTTP 1.1 GET request to determine if a new version is available

- **Login**

- **User Search**

- **Call Establishment and Teardown**

- **Media Transfer and Codec**

- **Keep-alive Messages**

- The SC sent a refresh message to its SN over TCP every 60s



Login

- **Login is perhaps the most critical function to the Skype operation**
- **During this process, a SC**
 - Authenticates its user name and password with the login server
 - Advertises its presence to other peers and its buddies
 - Determines the type of NAT and firewall it is behind
 - Discovers online Skype nodes with public IP addresses

Login Server and Bootstrap Super Nodes

- **Login Server**

- The only central component in the Skype network
- IP address: 80.160.91.11
 - ns14.inet.tele.dk and ns15.inet.tele.dk

- **Bootstrap Super Nodes**

- HC was initialized with 7 IP:Port pairs
- Bootstrap SNs are connected to the Internet through 4 ISPs
- If the HC was flushed after the first login, SC was unable to connect to the Skype Network

First-time Login Process [1/2]

- **There are only 7 entries in the SC host cache upon installation**
- **A SC must connect to well known Skype nodes in order to log on to the Skype Network**
 - By sending UDP packets to some bootstrap SNs and then wait for their response
 - It is not clear how SC selects among bootstrap SNs to send UDP packets to
 - SC then established a TCP connection with the bootstrap SN that responded



First-time Login Process [2/2]

- **A SC running on a machine with public IP address**
 - Exchange some packets with SN over TCP
 - Then establishes a TCP connection with the login server
 - The TCP connection with the SN persisted as long as SN was alive
 - The total data is about 9k bytes
- **A SC behind a port-restricted NAT**
 - Roughly the same as for a SC on a public IP address
 - The total data is about 10k bytes
- **A SC behind a port-restricted NAT and UDP-restricted firewall**
 - Unable to receive any UDP packets from machines outside the firewall
 - It exchanged 8.5k bytes of data



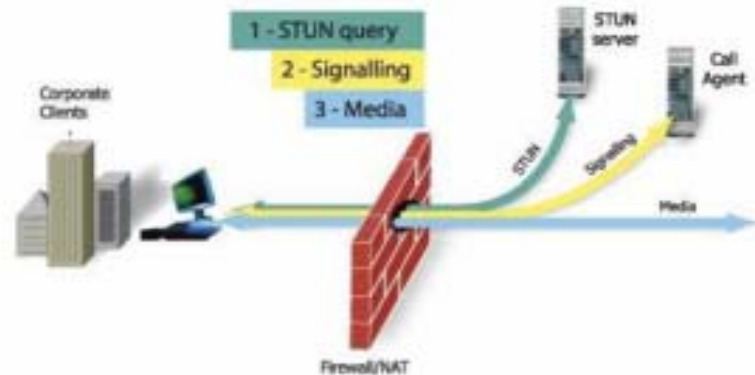
NAT and Firewall Determination

- **The authors conjecture that a SC is able to determine at login if it is behind a NAT and firewall**
 - By exchanging messages with **its SN** or **some nodes** using a variant of the STUN protocol
- **Once determined, the SC stores this information in the Windows registry**
- **SC refreshes this information periodically**

STUN and TURN

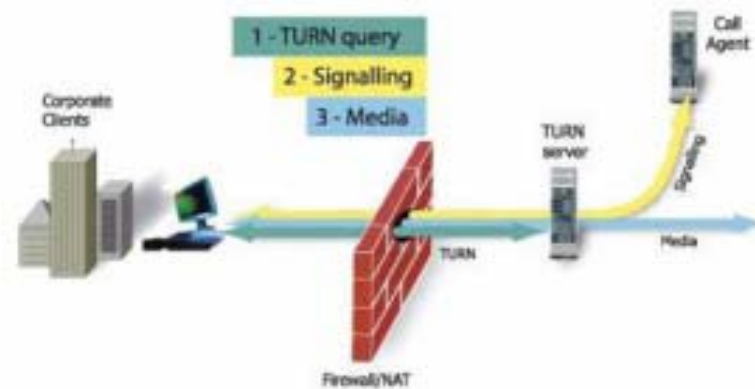
- **STUN**

- Simple Traversal of UDP through NAT
- Doesn't work through symmetric NAT



- **TURN**

- Traversal Using Relay NAT
- Increase latency
- Server load



Login Procedures

- **Alternate Node Table**
 - **SC sends UDP packets to about 20 distinct nodes at the end of login process**
 - To advertise its arrival on the network
 - **Upon receiving a response from them, SC builds a table of online nodes**
 - Alternate node table
 - It is with these nodes a SC can connect to, if its SN becomes unavailable
- **Subsequent Login Process**
 - Quite similar to the first-time login process
- **Login Process Time**
 - Scenario (1) and (2): **3-7** seconds
 - Scenario (3): about **34** seconds



User Search

- **Skype uses its Global Index (GI) technology to search for user**
 - A distributed algorithm
 - Guarantee to find a user if it exists and has logged in during the last 72 hours
- **For SC on a public IP address**
 - SC sent a TCP packet to its SN
 - SN gave SC the IP:Port of **4** nodes to query
 - If it could not find the user, it informed the SN over TCP
 - It appears that the SN now asked it to contact **8** different nodes
 - This process continued until the SC found the user or it determined that the user did not exist
 - The search took 3 to 4 seconds
- **Search Result Caching**



Call Establishment and Teardown [1/2]

- **The call signaling is always carried over TCP**
- **For users that are not in the buddy list**
 - Call placement = user search + call signaling
- **Both users were on public IP address**
 - The caller SC established a TCP connection with the callee SC
- **The caller was behind port-restricted NAT and callee was on public IP address**
 - The caller sent signaling information over TCP to an online Skype node which forwarded it to callee over TCP
 - The online node also routed voice packets from caller to callee over UDP and vice versa



Call Establishment and Teardown [2/2]

- **Both users were behind port-restricted NAT and UDP-restricted firewall**
 - Caller SC sent media over TCP to an online node, which forwarded it to callee SC over TCP and vice versa
- **Advantages of having a node route the voice packets from caller and callee**
 - It provides a mechanism for users behind NAT and firewall to talk to each other
 - If other users want to participate in a conference, this node serves as a mixer
- **Call tear-down**



Media Transfer and Codecs [1/2]

- **The total uplink and downlink bandwidth used for voice traffic is 5k bytes/s**
 - This bandwidth usage corresponds with the Skype claim of 3k-16k bytes/s
- **No silence suppression is supported in Skype**
 - It maintains the UDP bindings at NAT
 - These packets can be used to play some background noise at the peer
- **Skype allows peers to hold a call**
 - To ensure UDP binding, a SC sends three UDP packets per second to the call peer on average



Media Transfer and Codecs [2/2]

- **Codec Frequency Range**

- The min. and max. audible frequency Skype codecs allow to pass through are 50 Hz and 8000 Hz

- **Congestion**

- Uplink and downlink bandwidth of **2k** bytes/s each was necessary for reasonable call quality
- The voice was almost unintelligible at an uplink and downlink bandwidth of **1.5k** bytes/s

Conferencing

- A acts as a mixer, mixing its own packets with those of B and sending to C and vice versa
 - The most powerful machine will be elected as conference host and mixer
- Two-way call: 36k bytes/s
- Three-user conference: 54k bytes/s

