Overlay and P2P Networks

Unstructured networks

Prof. Sasu Tarkoma

16.1.2013
Contents

• Terminology and overlays continued

• Unstructured networks
  – Today
    – Napster
    – Skype
  – Next week:
    – Gnutella
    – BitTorrent
    – Freenet

• Summary
Terminology

• Peer-to-peer (P2P)
  – Different from client-server model
  – Each peer has both client/server features
• Overlay networks
  – Routing systems that run on top of another network, such as the Internet.
• Distributed Hash Tables (DHT)
  – An algorithm for creating efficient distributed hash tables (lookup structures)
  – Used to implement overlay networks
• Typical features of P2P / overlays
  – Scalability, resilience, high availability, and they tolerate frequent peer connections and disconnections
Peer-to-peer in more detail

• A P2P system is distributed
  – No centralized control
  – Nodes are symmetric in functionality

• Large faction of nodes are unreliable
  – Nodes come and go

• P2P enabled by evolution in data communications and technology

• Current challenges:
  – Security (zombie networks, trojans), IPR issues

• P2P systems are decentralized overlays
Characteristics of P2P systems

P2P can be seen as an organizational principle
   Applied in many different application domains

Characteristics
   Self-organization
   Lack of central coordination
   Resource sharing
   Based on collaboration between peers
   Peers are typically equal
   Large number of peers
   Resilient to certain kinds of attacks (but vulnerable to others)
P2P Volume

Estimates range from 10-20% of Internet Traffic
Sandvine Nov 2013 report: 10% (was 60% in 2001 and 2007 31%)

Latest estimates from Cisco suggest that video delivery is the growing and the share of P2P file exchange traffic is becoming smaller

P2P can be used for video delivery as well

.. And voice (Skype, P2PSIP)

Hundreds of millions of people use P2P technology today
Evolution of P2P systems

- **ARPAnet** had P2P like qualities
  - End-to-end communication, FTP, USENET,..
  - Today’s BGP is P2P
- Started from centralized servers
  - **Napster**
    - Centralized directory
    - Single point of failure
- Second generation used flooding (**Gnutella v0.4**)
  - Local directory for each peer
  - High cost, worst-case O(N) messages for lookup
  - Third generation use some structure (**Gnutella v0.7**)
- Research systems use **DHTs**
  - Chord, Tapestry, CAN, ..
  - Decentralization, scalability
- Some recent **CDNs** and content delivery systems exhibit P2P features (P2P assisted CDN)
Unstructured networks

Unstructured networks are typically based on random graphs following flat or hierarchical organization.

Unstructured networks utilize flooding and similar opportunistic techniques, such as random walks, expanding-ring, Time-to-Live (TTL) search, in order to locate peers that have interesting data items.

Many P2P systems: Gnutella, Freenet, BitTorrent, …
Napster

Napster was a centralized P2P music sharing service (mp3s)

Launched in 1999 and made P2P popular and dubious from the legal viewpoint
Lawsuits from 1999, close-down in 2001, Chapter 7 in 2002, rebirth as a music store in 2003

Utilized a centralized index (server farm) for searching, transfers were peer-to-peer
User installing the software
  Download the client program
  Register name, password, local directory, etc.
1. Client contacts Napster (via TCP)
  Provides a list of music files it will share
  … and Napster’s central server updates the directory
2. Client searches on a title or performer
  Napster identifies online clients with the file
  … and provides IP addresses
3. Client requests the file from the chosen supplier
  Supplier transmits the file to the client
  Both client and supplier report status to Napster
Napster Summary

Centralized server allows
  Consistent view of the P2P network
  Search guaranteed to find all files in the network

Limitations of this design are
  Centralized server is the weakest point of the system
    Attacks, network partitions, …
  Limited scalability
Skype

Skype is a well-known Internet telephony service
Calls between peers
Interface to traditional telephony services (costs money)

Skype architecture is similar to KaZaa and Gnutella
Supernodes and regular nodes
Developed by makers of Kazaa, now owned by Microsoft
A proprietary protocol, protocol uses encryption
A centralized server for logging and billing
Supernodes and regular nodes maintain a distributed directory of online peers
Supernodes forward calls and call traffic (mostly for firewalled/natted peers)
A number of built-in techniques for traversing firewalls and NAT boxes, STUN-like behaviour
What is NAT

Expand IP address space by deploying private address and translating them into publicly registered addresses

Private address space (RFC 1918, first in RFC 1631)

10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
192.168.0.0 - 192.168.255.255 (192.168.0.0/16)

Technique of rewriting IP addresses in headers and application data streams according to a defined policy
Based on traffic source and/or destination IP address

Source: Tanenbaum 4th
NAT Traversal

Main problem: how to reach a client with a private IP address?
Depends on the type of NATs and their mapping behaviour

Techniques
- Relaying
- Connection reversal
- Hole punching

Standards
- TURN
- STUN
- ICE

See additional material
Hole Punching with Restricted Cone

NAT routes the packet to the rendezvous who acquires the public address ip:port. A mapping is created.

Host looks up the public address of the device (NAT’s IP)

Relay with public IP address

Natted host sends message to host. Hole is punched

Private IP address

Public IP address
NATs and Firewalls

Firewalls
- Security main concern
- Demilitarized zone
- Increasingly complex rules (what is filtered, how)

NATs
- Lightweight security devices
  - Topology hiding and firewalling
- Increasing number in deployment
  - Solves some of the address space problems of IPv4 (Port Translation, NAPT)
- IPv6 solves the addressing problem so NATs are not needed for this
Skype

- Skype is P2P
- Proprietary application-layer protocol
- Hierarchical overlay with super nodes
- Index maps usernames to IP addresses; distributed over super nodes
- Peers with connectivity issues use NAT traversal or communicate via super node relays
- Developer API

- Security: RSA, AES for voice, RC4 obfuscation for payload, authentication with Skype Servers
Problem when both Alice and Bob are behind "NATs".

NAT prevents an outside peer from initiating a call to insider peer

Solution:

Using Alice’s and Bob’s SNs, Relay is chosen

Each peer initiates session with relay.

Peers can now communicate through NATs via relay
User Search

Skype uses a global index to search for a user

UDP/TCP between Skype nodes and/or super nodes

Skype claims that search is distributed and is guaranteed to find a user if it exists and has logged in during last 72 hours

Search results are observed to be cached at intermediate nodes
Login

1. Login routed through a super node. Find super nodes by sending UDP packets to bootstrap super nodes (defaults) and wait for responses
2. Establish TCP connections with selected super nodes based on responses
3. Acquire the address of a login server and authenticate user
4. Send UDP packets to a preset number of nodes to advertise presence (a backup connectivity list).

Host Cache (HC) is a list of super node IP address and port pairs that Skype Client maintains.
Login algorithm (HC is host cache, Login server not shown)

Establishing a Call: Three cases

Case 1: Public IP addresses. Caller establishes TCP connection with callee Skype client.

Case 2: Caller is behind port-restricted NAT, callee has public IP. Caller uses online Skype node to forward packets over TCP/UDP.

Case 3: Both caller and callee behind port-restricted NAT and UDP restricted firewall. Exchange info with a Skype node using TCP. Caller sends media over TCP to an online node which forwards to callee via TCP.

Port-restricted NAT: An external host can send a packet, with source IP address $X$ and source port $P$, to the internal host only if the internal host had previously sent a packet to IP address $X$ and port $P$. 
Signalling

The Skype client will use UDP for voice if it is behind a NAT or firewall that allows UDP packets to flow across TCP is used for signalling

Media is always transferred with UDP unless both caller and callee are behind port-restricted NAT and UDP-restricted firewall
Skype and NATs

Comparison of three network setups
Exp A: both Skype users with public IP address
  Users are online and on each other’s buddy lists
Exp B: Skype caller/callee behind port-restricted NAT (incoming port must be the one that sent the packet, more difficult to punch a hole). **One super node in use.**
Exp C: Both Skype users behind port-restricted NAT and UDP-restricted firewall. **Multiple super nodes.**

Message flows for first time login process
Exp A and Exp B are similar
Exp C only exchange data over TCP

<table>
<thead>
<tr>
<th></th>
<th>Total data exchanged</th>
<th>Login process time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp A</td>
<td>Approx 9 KB</td>
<td>3-7 secs</td>
</tr>
<tr>
<td>Exp B</td>
<td>Approx 10 KB</td>
<td>3-7 secs</td>
</tr>
<tr>
<td>Exp C</td>
<td>Approx 8.5 KB</td>
<td>Approx 34 secs</td>
</tr>
</tbody>
</table>

Skype and NATs

Skype uses a variation of STUN and TURN

The Skype client attempts to identify the NAT type during the login phase

Super nodes are relay servers
Skype Security Basics

All communications encrypted (symmetric session key and AES, bootstrap with public key crypto, RC4 for obfuscation)

1. Client authenticates with login server (public key crypto)
2. Login server issues a certificate for client’s public key
3. Client certificate is disseminated to supernodes
4. Certificate is returned if someone searches for the user
5. Public key crypto is used to exchange sessions keys
Security: Details

1. Skype client has a built in list of Skype login servers and their public keys (Ks+)
2. Users first register username and a hash of password (H(pwd)) at the server (encrypt with server public key)
3. On each login session, Skype client generates a session key K
4. Skype client also generates a 1024-bit private/public RSA key pair (KA+, KA-).
5. Skype client sends Ks+ (K), K (KA+, Username, H(pwd)) to server and obtains a certificate for the Username, public key pair (only if password is valid)
6. Certificate is disseminated to Super Nodes
7. Skype clients can then authenticate by exchanging certificates and verifying that a Skype server has signed the certificates
8. Final step is to derive a session key that is used to encrypt all communications
Blocking skype

Skype traffic looks suspicious
  Encrypted, traffic even if no calls or activity
  Moreover, code is obfuscated

Skype binary in 2006 had
  binary packing, code integrity, anti-debug, obfuscation

Firewall rules

Skype traffic detection

http://www.tml.tkk.fi/Publications/C/23/papers/
  Santolalla_final.pdf
Supernode map (Infocom 2006 article)
Simultaneous signed in Skype users

High-water marks: *Signed In Skype Accounts* over time

Skype sign-ups swell **October through March 2011**, adding **6 million** concurrent users, followed by seven months without growth. The season averaged **38.7K** more signed-in users daily **+25%**

Little Seasonality at the start

Huge Seasonality Now

2012's **104-day 9 million spike** didn't start until January, averaging **95.5K** more signed-in users daily.

@evanwolf, Phil Wolff, Skype Journal, 23 April 2012
Skype vulnerability to blackouts I/II

Skype has had a number of blackouts
In 2007 Microsoft Windows Update caused a blackout
• High number of reboots reduced the number of super nodes in operation
• The number of super nodes was not sufficient to handle the load
• http://heartbeat.skype.com/2007/08/the_microsoft_connection_explained.html
Skype vulnerability to blackouts II

One of the most severe was in December 2010

- Loss of 10 million calls
- Users unable to connect to super nodes

- Bug in Skype client $\rightarrow$ 40% clients fail $\rightarrow$ 25-30% supernodes fail $\rightarrow$ overload $\rightarrow$ feedback loop shutting down overloaded supernodes $\rightarrow$ global blackout

- Fix: Skype engineers start more supernodes
Skype today

Skype was acquired by Microsoft in 2011-2012

Changes in 2012

Number of supernodes went from 48k to 10k
Supernodes hosted by Microsoft in datacenters
It is not possible for a regular node to be elevated to a supernode
Privacy concerns
Conclusion on Skype

Successful overlay and P2P technology

Call forwarding with self-organizing network of nodes

Still P2P based; however, backed by an infrastructure supported super nodes