Scalable overlay Networks

Dr. Samu Varjonen
18.01.2018
Lectures

- **MO 15.01. C122 Introduction. Exercises. Motivation.**
- **TH 18.01. DK117 Unstructured networks I**
  - MO 22.01. C122 Unstructured networks II
  - TH 25.01. DK117 Bittorrent and evaluation
- **MO 29.01. C122 Privacy (Freenet etc.) and intro to power-law networks.**
- **TH 01.02. DK117 Consistent hashing. Distributed Hash Tables (DHTs)**
- **MO 05.02. C122 DHTs continued**
- **TH 08.02. DK117 Power-law networks**
- **MO 12.02. C122 Power-law networks and applications.**
- **TH 15.02. DK117 Applications I**
- **MO 19.02. C122 Applications I**
- **TH 22.02. DK117 Advanced topics**
- **MO 26.02. C122 Conclusions and summary**
- **TH 01.03. DK117 Reserved**
Contents

• Terminology and overlays continued

• Unstructured networks
  – Today
    • Napster
    • Skype

• Next week:
  • Gnutella
  • BitTorrent
  • Freenet

• Summary
Evolution of the network

- Video delivery is one of the services on the Web

- Global IP traffic has increased more than fivefold in the past 5 years and it will increase threefold in the next 5 years.

- CDNs will carry over half of Internet traffic in 2018. 55% of all Internet traffic will cross CDNs in 2018.

- Traffic from mobile and wireless devices will surpass wired traffic by 2018

- P2P share of the network is diminishing (around 10%)
### Cisco forecast

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>65,942</td>
<td>83,371</td>
<td>102,906</td>
<td>127,008</td>
<td>155,121</td>
<td>187,386</td>
<td>23%</td>
</tr>
<tr>
<td>Managed</td>
<td>22,911</td>
<td>27,140</td>
<td>31,304</td>
<td>35,226</td>
<td>38,908</td>
<td>42,452</td>
<td>13%</td>
</tr>
<tr>
<td>Mobile</td>
<td>7,201</td>
<td>11,183</td>
<td>16,646</td>
<td>24,220</td>
<td>34,382</td>
<td>48,270</td>
<td>46%</td>
</tr>
<tr>
<td>Total</td>
<td>96,054</td>
<td>121,694</td>
<td>150,910</td>
<td>186,453</td>
<td>228,411</td>
<td>278,108</td>
<td>24%</td>
</tr>
</tbody>
</table>

**Compound Annual Growth Rate (CAGR)**

\[
CAGR = \left( \frac{\text{Ending Value}}{\text{Beginning Value}} \right)^{\frac{1}{\# \text{ of years}}} - 1
\]

Managed IP traffic is limited to IP traffic that is managed from origin to destination by a single service provider, crossing only a single network.

Busy hour

- Busy hour Internet traffic is growing more rapidly than average Internet traffic.

- Busy-hour (or the busiest 60 minute period in a day) Internet traffic increased 51 percent in 2016, compared with 32-percent growth in average traffic.

- Busy-hour Internet traffic will increase by a factor of 4.6 between 2016 and 2021, while average Internet traffic will increase by a factor of 3.2.

Video

• It would take an individual more than 5 million years to watch the amount of video that will cross global IP networks each month in 2021.

• Internet video surveillance traffic increased 72 percent in 2016, from 516 Petabytes (PB) per month at the end of 2015 to 883 PB per month in 2016. Internet video surveillance traffic will increase sevenfold between 2016 and 2021.

• Virtual reality and augmented reality traffic will increase 20-fold between 2016 and 2021.

• Consumer Video-on-Demand (VoD) traffic will nearly double by 2021.

• Content Delivery Network (CDN) traffic will carry 71 percent of all Internet traffic by 2021.

Summary on growth

- Much of this increase comes from the delivery of video data

- P2P traffic has become a smaller component of Internet traffic in terms of its current share

- Video is being delivered by a set of protocols, typically coordinated by overlay solutions and CDN solutions

- We will cover these in during the course
CDN

- Content Delivery Networks (CDNs) are examples of overlay networks that cache and store content and allow efficient and less costly way to distribute data in massive scale.

- CDNs typically do not require changes to end-systems and they are not peer-to-peer solutions from the viewpoint of the end clients.
## Traffic trends, challenges and solutions

<table>
<thead>
<tr>
<th>Trend</th>
<th>Challenges</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2P</td>
<td>Growth in traffic, upstream bottlenecks</td>
<td>P2P caching, locality-awareness</td>
</tr>
<tr>
<td>Internet Broadcast</td>
<td>Flash crowds</td>
<td>P2P content distribution, multicast technologies</td>
</tr>
<tr>
<td>Internet Video-on-Demand</td>
<td>Growth in traffic, especially metropolitan area and core</td>
<td>Content Delivery Networks (CDNs), increasing network capacity, compression</td>
</tr>
<tr>
<td>Commercial Video-on-Demand</td>
<td>Growth in traffic in the metropolitan area network</td>
<td>CDNs, increasing network capacity, compression</td>
</tr>
<tr>
<td>High-definition content</td>
<td>Access network IPTV bottleneck, growth in VoD traffic volume in the metropolitan area network</td>
<td>CDNs, increasing network capacity, compression</td>
</tr>
</tbody>
</table>
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
Challenges for overlays

- **The Real World:** In practice, the typical underlay protocol, IP, does not provide universal end-to-end connectivity due to the ubiquitous nature of firewalls and Network Address Translation (NAT) devices.

- **Management and administration:** Practical deployment requires that the overlay network has a management interface.

- **Overhead:** An overlay network typically consists of a heterogeneous body of devices across the Internet. It is clear that the overlay network cannot be as efficient as the dedicated routers in processing packets and messages. Moreover, the overlay network may not have adequate information about the Internet topology to properly optimize routing processes.
Network address translation

- 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
NAT Traversal

• Challenge: how to allow two natted hosts communicate?

• Straightforward solution: use a relay with a public address that is not natted
  – Connection reversal possible if a node has a public address
  – Relay is a rendezvous point

• More complicated solutions
  – Detect presence of NATs
  – Hole punching

• Standards: STUN, TURN, ICE
Interactive Connectivity Establishment (ICE)

- With asymmetric NAT, ICE will use a STUN (Session Traversal Utilities for NAT) server. A STUN server allows clients to discover their public IP address and the type of NAT they are behind.
- If a STUN server cannot establish the connection, ICE can turn to TURN.
- Traversal Using Relay NAT (TURN) is an extension to STUN that allows media traversal over a NAT that does not do the “consistent hole punch” required by STUN traffic.
- TURN servers are often used in the case of a symmetric NAT.
- A TURN server remains in the media path after the connection has been established. That is why the term “relay” is used to define TURN.
Connection reversal

Private IP address

Public IP address

Relay with public IP address

B with public address uses relay to send the address to NATed node
Translation methods

• Full-cone NAT, also known as one-to-one NAT
  - Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are sent through eAddr:ePort.
  - Any external host can send packets to iAddr:iPort by sending packets to eAddr:ePort.

• (Address)-restricted-cone NAT
  - Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are sent through eAddr:ePort.
  - An external host (hAddr:any) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:any. "Any" means the port number doesn't matter.

• Port-restricted cone NAT
  - Like an address restricted cone NAT, but the restriction includes port numbers.
  - Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are sent through eAddr:ePort.
  - An external host (hAddr:hPort) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:hPort.

• Symmetric NAT
  - Each request from the same internal IP address and port to a specific destination IP address and port is mapped to a unique external source IP address and port; if the same internal host sends a packet even with the same source address and port but to a different destination, a different mapping is used.
  - Only an external host that receives a packet from an internal host can send a packet back.

https://www.usenix.org/legacy/event/usenix05/tech/general/full_papers/ford/ford.pdf
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
Network invariants and metrics

• The **correctness** and **performance** of a routing algorithm can be analyzed using a number of **metrics**

• Typically it is expected that a routing algorithm satisfies certain invariant properties that must be satisfied at all times. The two key properties are **safety** and **liveness**

• The former states that undesired effects do not occur, in other words the algorithm works correctly, and the latter states that the algorithm continues to work correctly, for example avoids deadlocks and loops

• These properties can typically be proven for a given routing algorithm under certain assumptions

• Important metrics: **shortest path**, **routing table size**, **path stretch**, **forwarding load**, **churn**
Example: Resilient Overlay Network

- Consider a native link failure in CE
- Only overlay link AE is affected.
- The native path AE is rerouted over F (ACE → ACFDE)
Overlay applications

• Typical applications of overlay networks include:
  - Content search and file transfer.
  - Distributed directories with efficient lookups.
  - Content routing over the Internet including voice and video.
  - Publish/subscribe and notification.
  - Distributed storage systems.
  - Multi-player games.

• Newer applications
  - NoSQL systems (Cassandra, …)
  - BigData processing
  - Control plane of SDN
P2P 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

- 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

- Includes traffic from P2P applications such as BitTorrent and eDonkey, as well as web-based file sharing. Note that a large portion of P2P traffic is due to the exchange of video files.

- Latest estimates from Cisco suggest that video delivery is the growing and the share of P2P file exchange traffic is becoming smaller (Cisco forecast CAGR -6%).

- Still hundreds of millions of people use P2P technology today.
Evolution of P2P

- 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

- Typically based on random graphs following flat or hierarchical organization

- 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
Napster

User installing the software

1. Client contacts Napster (via TCP)
   Download the client program
   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
Skype

- Skype is P2P (was?)
- 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
Skype peers as relays

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.
Calling: 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 a 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.
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 calle 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 & 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
Security in Skype

• RSA
  - One of the first practical public-key cryptosystems publicly described in 1977 by Rivest, Shamir and Adleman at MIT
  - Recipient's public key is used for encryption
  - Sender’s private key is used for digital signatures
  - Skype uses RSA for authentication and exchange of symmetric keys

• AES (Advanced Encryption Standard) is a specification established by NIST in 2001 based on the Rijndael cipher.
  - Symmetric key algorithm
  - Skype uses AES for protecting the exchange of key material with the login server and for encrypting voice traffic

• RC4
  - Most widely used stream cipher used for example in TLS (as of 2015 prohibited)
  - Possible vulnerabilities
  - Skype uses RC4 to obfuscate signalling.
Skype security basics

• All communications are encrypted

• Client authenticates with login server using public key crypto

• Login server issues a certificate for client’s public key

• Client certificate is disseminated to supernodes

• Certificate is returned if someone searches for the user

• Public key crypto is used to exchange symmetric 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 with the client RSA key pair that is used to encrypt all communications.
Blocking Skype

• Skype traffic looks suspicious
  – Encrypted, traffic even if no calls or activity

• Code is obfuscated. Skype binary in 2006 had binary packing, code integrity, anti-debug, obfuscation

• Firewall rules

• Skype traffic detection
  – Naive Bayes classifiers and other techniques

Supernode map

Infocom 2006 article:
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
Blackouts II / II

- One of the most severe was in December 2010
  - Loss of 10 million calls
  - Users unable to connect to super nodes

- What happened:
  - Bug in Skype client
    - -> 40% clients fail
    - -> 25-30% supernodes fail
    - -> overload
      - -> feedback loop shutting down overloaded supernodes
      - -> global blackout

- Fix: Skype engineers start more supernodes

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, call routing is not P2P anymore

• It is not possible for a regular node to be elevated to a supernode

• Privacy concerns
Conclusion

• One of the first successful overlay and P2P technologies

• Call forwarding with self-organizing network of nodes

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