Contents

• Terminology and overlays continued

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

• Summary
**Evolution of the network**

**Video delivery** has become one of the recent 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: Global Video Highlights

**Internet video** to TV doubled in 2012 and will increase fivefold by 2017. Internet video to TV traffic will be 14% of consumer Internet video traffic in 2017, up from 9 percent in 2012.

**Video-on-demand** traffic will nearly triple by 2017. The amount of VoD traffic in 2017 will be equivalent to 6 billion DVDs per month.

**High-definition** video-on-demand surpassed standard-definition VoD by the end of 2011.
Global IP traffic

\[
\text{CAGR} = \text{Compounded Annual Growth Rate} \\
\text{CAGR}(t_0, t_n) = \left(\frac{V(t_n)}{V(t_0)}\right)^\frac{1}{t_n-t_0} - 1
\]

Exponential growth

21% for 2013-2018

Source: Cisco VNI Mobile Forecast, 2013
## Consumer traffic

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>By Network (PB per Month)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>By Subsegment (PB per Month)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet video</td>
<td>0.52</td>
<td>1.07</td>
<td>2.03</td>
<td>3.14</td>
<td>4.34</td>
<td>5.74</td>
<td>29%</td>
</tr>
<tr>
<td>Web, email, and data</td>
<td>5.17</td>
<td>6.34</td>
<td>7.78</td>
<td>9.54</td>
<td>11.83</td>
<td>14.49</td>
<td>23%</td>
</tr>
<tr>
<td>File sharing</td>
<td>6.20</td>
<td>7.12</td>
<td>7.82</td>
<td>8.27</td>
<td>8.48</td>
<td>8.67</td>
<td>7%</td>
</tr>
<tr>
<td>Online gaming</td>
<td>22</td>
<td>26</td>
<td>32</td>
<td>39</td>
<td>48</td>
<td>59</td>
<td>22%</td>
</tr>
<tr>
<td><strong>By Geography (PB per Month)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>2,194</td>
<td>2,757</td>
<td>3,433</td>
<td>4,182</td>
<td>5,015</td>
<td>5,897</td>
<td>22%</td>
</tr>
<tr>
<td>North America</td>
<td>312</td>
<td>14,188</td>
<td>17,740</td>
<td>21,764</td>
<td></td>
<td></td>
<td>26%</td>
</tr>
<tr>
<td>Western Europe</td>
<td>410</td>
<td>640</td>
<td>944</td>
<td>1,334</td>
<td>1,816</td>
<td>2,432</td>
<td>43%</td>
</tr>
<tr>
<td>Central and Eastern Europe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latin America</td>
<td>2,656</td>
<td>3,382</td>
<td>4,049</td>
<td>4,588</td>
<td>5,045</td>
<td>5,487</td>
<td>16%</td>
</tr>
<tr>
<td>Middle East and Africa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total (PB per Month)</strong></td>
<td>26,213</td>
<td>33,337</td>
<td>41,429</td>
<td>50,809</td>
<td>62,269</td>
<td>75,973</td>
<td>24%</td>
</tr>
</tbody>
</table>

Source: Cisco VNI, 2013
Dynamic Patterns

**Busy hour Internet traffic** is growing more rapidly than average Internet traffic. Busy hour Internet traffic increased 41 percent in 2012, compared to 34 percent growth in average traffic.

Busy-hour Internet traffic will increase by a factor of 3.5 between 2012 and 2017, while average Internet traffic will increase 2.9-fold.

Busy-hour Internet traffic will reach 865 Tbps in 2017, the equivalent of 720 million people streaming a high-definition video continuously.
Network 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.
CDNs

*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.
<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 Overlay Networks

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 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 Overlay: RON

- Consider a native link failure in $CE$
- Only overlay link $AE$ is affected.
- The native path $AE$ is rerouted over $F$ ($ACE \rightarrow ACFDE$)
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
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 2014 report 8% (was 10% in 2013, 31% in 2007, 60% in 2001)

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

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
Connection reversal

B with public address uses relay to send the address to NATed node
Hole Punching with Restricted Cone

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

Private IP address

Natted host sends message to host. Hole is punched

Public IP address

Relay with public IP address

Restricts traffic based on public IP address (not on port)

(X,y) sends to (A,z) through (N,q)

(A,w) can send back to (N,q)
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)

Start

- Send UDP packet(s) to HC IP address and port

  - Response within 5 seconds
    - Yes
      - Connected
      - Yes
      - Success
    - No
      - TCP connection attempt with HC IP address and port
        - Connected
          - Yes
          - Connected
          - Yes
          - Success
        - No
        - TCP connection attempt with HC IP address and port 80 (HTTP port)
          - Connected
          - Yes
          - Connected
          - Yes
          - Success
          - No
          - Connection Attempts == 5
            - Yes
            - Failure
            - No
            - Wait for 6 seconds

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 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.

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><strong>Exp A</strong></td>
<td>Approx 9 KB</td>
<td>3-7 secs</td>
</tr>
<tr>
<td><strong>Exp B</strong></td>
<td>Approx 10 KB</td>
<td>3-7 secs</td>
</tr>
<tr>
<td><strong>Exp C</strong></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: Building Blocks

RSA
One of the first practical public-key cryptosystems publicly described in 1977 by Rivest, Shamir and Adleman at MIT
Recipient's public 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.
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
Possible vulnerabilities
Skype uses RC4 to obfuscate signalling.
Skype Security Basics

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

1. Client authenticates with login server (public key crypto)

1. Login server issues a certificate for client’s public key

1. Client certificate is disseminated to supernodes

1. Certificate is returned if someone searches for the user

1. 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 \( (K_{s+}) \)
2. Users first register username and a hash of password \( (H(pwd)) \) at the server \( (\text{encrypt with server public key}) \)
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 \( K_{s+} (K), K (KA+, \text{Username}, H(pwd)) \) 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)
Simultaneous signed in Skype users

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%

Huge Seasonality Now

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

Little Seasonality at the start

@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 → 40% clients fail → 25-30% supernodes fail → overload → feedback loop shutting down overloaded supernodes → 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, call routing is not P2P anymore

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