Lecture Plan

• Lecture
• Overview for final
• Class reviews
P2P dominates Internet traffic

(1) Source: CacheLogic “P2P in 2005,” (9/05).
What is a P2P System?

• A system leveraging client resources to attain scalability, performance, reliability, etc.

• In contrast to the client-server paradigm, P2P systems generally have few centralized components
Why are P2P systems used?

- **Scalability** (BitTorrent)
  - A centralized server does not scale well for large file distribution. Make each peer a source of content.

- **Resilience** (RON) and Anonymity (Tor\Freenet)
  - Can detect Internet outages and route around them through peers. Can also route through peers to avoid monitoring.

- **Performance** (Skype)
  - Inter-domain routing (BGP) is optimized for cost, not latency. Route through peers for lower latency, or better throughput.

- **Reliability** (Amazon’s S3, Volunteer Computing with BOINC)
  - Network outages are strongly correlated. Don’t place all your eggs in one basket. 10 Independent answers to a simulation are better than 10 answers from the same host.
Why are P2P systems used?

- **Scalability** *(BitTorrent)*
  - A centralized server does not scale well for large file distribution. Make each peer a source of content.

- **Resilience** *(RON)* and **Anonymity** *(Tor\Freenet)*
  - Can detect Internet outages and route around them through peers. Can also route through peers to avoid monitoring.

- **Performance** *(Skype)*
  - Inter-domain routing (BGP) is optimized for cost, not latency. Route through peers for lower latency, or better throughput.

- **Reliability** *(Amazon’s S3, Volunteer Computing with BOINC)*
  - Network outages are strongly correlated. Don’t place all your eggs in one basket. 10 Independent answers to a simulation are better than 10 answers from the same host.
Why are P2P systems used?

- **Scalability** (*BitTorrent*)
  - A centralized server does not scale well for large file distribution. Make each peer a source of content.

- **Resilience** (*RON*) and **Anonymity** (*Tor*/*Freenet*)
  - Can detect Internet outages and route around them through peers. Can also route through peers to avoid monitoring.

- **Performance** (*Skype*)
  - Inter-domain routing (*BGP*) is optimized for cost, not latency. Route through peers for lower latency, or better throughput.

- **Reliability** (Amazon’s *S3*, Volunteer Computing with *BOINC*)
  - Network outages are strongly correlated. Don’t place all your eggs in one basket. 10 Independent answers to a simulation are better than 10 answers from the same host.
Why are P2P systems used?

- **Scalability (BitTorrent)**
  - A centralized server does not scale well for large file distribution. Make each peer a source of content.

- **Resilience (RON) and Anonymity (Tor\Freenet)**
  - Can detect Internet outages and route around them through peers. Can also route through peers to avoid monitoring.

- **Performance (Skype)**
  - Inter-domain routing (BGP) is optimized for cost, not latency. Route through peers for lower latency, or better throughput.

- **Reliability (Amazon’s S3, Volunteer Computing with BOINC)**
  - Network outages are strongly correlated. Don’t place all your eggs in one basket. 10 Independent answers to a simulation are better than 10 answers from the same host.
P2P Properties : Challenges

• Decentralization:
  • Distributed organization of many nodes
  • Bootstrapping new nodes into the system
  • Distributed lookup - locating the right peers (for content, network proximity, etc)

• Non-dedicated resources:
  • Nodes churn (join and leave the system) and fail randomly
  • Heterogeneity of peer resources (power-law distribution)
P2P Properties: Challenges

- No centralized vantage point on peer interactions:
- Incentivizing good behavior from independent actors
- Preventing peer highjacking and protocol abuse that may cause Distributed Denial of Service (DDoS) attacks
P2P Examples: Napster

• Not a true P2P system
• Uses a central index for
  • Finding content
  • Mapping content to peers
• File download happens between two peers
P2P Examples: Napster

- Not a true P2P system
- Uses a central index for
  - Finding content
  - Mapping content to peers
- File download happens between two peers
P2P Examples: Napster

• Not a true P2P system
• Uses a central index for
  • Finding content
  • Mapping content to peers
• File download happens between two peers
P2P Examples: Napster

- Not a true P2P system
- Uses a central index for
  - Finding content
  - Mapping content to peers
- File download happens between two peers
P2P Examples : Napster

• Not a true P2P system

• Uses a central index for
  • Finding content
  • Mapping content to peers

• File download happens between two peers

Hosts A,C,D
P2P Examples: Napster

- Not a true P2P system
- Uses a central index for
  - Finding content
  - Mapping content to peers
- File download happens between two peers
P2P Examples: Napster

- Not a true P2P system
- Uses a central index for
  - Finding content
  - Mapping content to peers
- File download happens between two peers
P2P Examples: Napster

- Not a true P2P system
- Uses a central index for finding content and mapping content to peers
- File download happens between two peers

What do you think are the challenges for Napster?
Gnutella

- Nodes must know another node in the system to join
- Content seekers send out \texttt{query}(qid, \texttt{ttl}) messages to neighbors
- Neighbors record the (qid, source IP) in a ‘routing table’, and flood \texttt{query}(qid, \texttt{ttl}-1) messages to neighbors
- Content sources respond with \texttt{query response}(qid) messages. These are routed back to content seekers by matching the qid in the local routing table.
- \texttt{Ping/Pong} heartbeat messages are used to probe existing nodes
A
Neighbors: B, F

Content Seeker

B

C

D

E

G

F

Content Source

Gnutella
Gnutella

Neighbors: B, F

Content Seeker

Content Source
Gnutella

Neighbors: B, F

Content Seeker

Q(1,2)

I : A

Content Source
Gnutella

Neighbors: B, F

Content Seeker

A

I : A

B

Q(1,1)

D

Q(1,1)

E

I : F

Content Source

G

I : F

Q(1,1)

F

I : A

Q(1,1)

Drop

C

I : B
Gnutella

Neighbors: B, F

Content Seeker

A

B

I : A

Q(1,1)

C

I : B

TTL = 0

D

E

I : F

F

I : A

Q(1,1)

G

I : F

Q(1,1)

Content Source

10
Gnutella

Neighbors: B, F

Content Seeker

A

B

C

D

E

F

G

QR(1)

I : A

I : B

I : A

I : F

Content Source

1 : F

1 : F

1 : A

1 : B
Gnutella

Neighbors: B, F, E

A

B

C

D

E

G

F

I : A

I : B

I : F

Content Seeker

Content Source
What do you think are the challenges for Gnutella?
M joins the system by obtaining a random subset of current peers from a centralized coordinator called a tracker.
BitTorrent overview
BitTorrent overview
BitTorrent overview
BitTorrent overview
BitTorrent overview

1 2 3 4

S

M

1 2 4

A

B

1 4
BitTorrent overview
BitTorrent overview
Evolution of P2P incentives

• Early P2P systems did not provide contribution incentives
  • 70% of Gnutella users didn’t share (free-riding)
  • 50% of queries answered by 1% of hosts

• Subsequent designs:
  • “Incentive priorities” in Kazaa were spoofed
  • Centralized accounting (MojoNation) not adopted

• **BitTorrent** is the most popular P2P system with explicit, decentralized contribution incentives
Why need incentives?
Because host capacity is heavily skewed

(Piatek et al. NSDI’07)
Why need incentives? Because host capacity is heavily skewed

65% of peer upload capacities ≤ 100 KBps

(Piatek et al. NSDI’07)
Tit-for-tat in BitTorrent

Choosing peers and rates:

1. Sort peers by incoming data rate

2. Reciprocate with top $k$, $k \propto \sqrt{\text{rate}}$

3. Send each peer selected in (2) an equal split of capacity

<table>
<thead>
<tr>
<th>Peer</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>17</td>
</tr>
<tr>
<td>C</td>
<td>13</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
</tr>
</tbody>
</table>

If $k=2$, P reciprocates with A and C, sending to each at an equal rate
Choosing peers and rates:

1. Sort peers by incoming data rate.

2. Reciprocate with top \( k \), \( k \propto \sqrt{\text{rate}} \).

3. Send each peer selected in (2) an equal split of capacity.

If \( k=2 \), P reciprocates with A and C, sending to each at an equal rate.