Peer to Peer networks
P2P example
P2P networks

- Alice wants the song “foo”
  - she turns on Limewire
- Bob has the P2P application turned on and “foo” in the shared folder
- Alice’s client finds out that “foo” is on Bob’s server
- Alice initiates a direct connection with Bob and downloads “foo”
P2P

• More traffic than any other application

• Mostly media content

• Multiple issues
  - Security
  - Privacy
  - Anonimity
  - Copyright Infringement
  - Intellectual property
Copyright infringement on P2P

Non-Infringing Intent of Use

LimeWire BASIC is a P2P program for use only in the exchange of authorized files. Downloading LimeWire BASIC does not constitute a license for obtaining or distributing unauthorized material. Please do not download LimeWire BASIC if you intend to use it to infringe copyright.

Find out more

- I might use LimeWire BASIC for copyright infringement.
- I will not use LimeWire BASIC for copyright infringement.

Continue >

~ Click Here for Important Information about Using P2P Software Safely ~

MERCHANDISE | PRESS | COMPANY | CONTACT | COPYRIGHT | PRIVACY
Peer-to-Peer (P2P) systems are increasingly becoming popular.

P2P file-sharing systems, such as Gnutella, Napster and Freenet realized a distributed infrastructure for sharing files.

Traditionally, files were shared using the Client-Server model (e.g. http). Not scalable since they are centralized services.

P2P uncover new advantages in simplicity of use, robustness, self organization and scalability.
P2P Information Retrieval

Problem:
“How to efficiently retrieve Information in P2P systems where each node shares a collection of documents?”

- Documents consists of keywords.
- Resembles Information Retrieval but resources are distributed now.
- Primary Data Structures such as Global Inverted Indexes can’t be maintained efficiently.
Why is this more difficult than centralized IR?
- Selection of nodes to query
  - who is up?
- Merging of results
- Spam
- Caching difficult; content changing fast
Peer-To-Peer (P2P) Search

- Distributed environment
  - Everybody does everything

- Each node in a network builds and maintains its own index

- Each node has “servent” software
  - On booting, servent pings ~4 other hosts
  - Connects to those that respond
  - Initiates, propagates and serves requests
Which hosts to connect to?

- The ones you connected to last time
- Random hosts you know of
- Request suggestions from central (or hierarchical) nameservers

- All govern system’s shape and efficiency
P2P networks

- **1st generation**
  - Centralized directory
  - Napster

- **2nd generation**
  - Gnutella

- **3rd generation**
  - FastTrack
  - Ultrapeers/SuperNodes
Figure 2.23  The P2P paradigm with a centralized directory
1\textsuperscript{st} generation P2P

- centralized server containing most of information on the network
  - File names mapped to IPs

- single point failure

- performance bottleneck

- copyright infringement easy to track
  - Napster shut down in 2000
2nd generation P2P

File transfer

Query -> Query hit

Query -> Query hit

Query -> Query hit

Query -> Query hit

Query -> Query hit

Query -> Query hit

Query -> Query hit

Figure 2.24  ♦  Search and file transfer in Gnutella
2nd generation P2P

- Gnutella
  - Limewire, Morpheus, BearShare etc

- Much harder to pursue in court

- Not-so-scalable
2nd generation P2P

1. Breadth-First Search (Gnutella)
   Query Flooding
   - Each Query Message is propagated along all outgoing links of a peer using TTL (time-to-live).
   - TTL is decremented on each forward until it becomes 0
   - Technique for I.R in P2P systems such as Gnutella.
   - Results?
     - The physical network comes to its knees
     - Long Delays for search results.

![Diagram showing a P2P network with an initial query/query hit exchange between peers](image-url)
Figure 2.25  ♦ Hierarchical overlay network for P2P file sharing
ultrapeers/supernodes

- status “ultrapeer” given by
  - uptime
  - bandwidth
  - number of downloads
  - neighbors
  - need etc.

- exchange most of the info; act like Gnutella within UltraPeers
- act like a Napster for their leaves
- very scalable
P2P - KazaA

- request queing
- incentive priorities
  - the more one uploads the better
- parallel downloading
- proprietary technology
  - encrypts all control traffic
  - numerous reverse engineering attempts
  - KazaA Lite
- hard for US organization to sue
  - patents held in Netherlands;
  - headquarters in Australia
  - developers in Estonia
Techniques for Distributed IR

Modified Random BFS

- Each Query Message is forwarded to only a fraction of outgoing links (e.g. $\alpha$ of them).
- TTL is again decremented on each forward until it becomes 0.
- Results?
  - Fewer Messages but possibly less results
  - This algorithm is probabilistic.
  - Some segments may become unreachable

Peer $q$

Peer $d$

P2P Network $N$
Techniques for Distributed I.R.

Intelligent Search Mechanism (ISM)

Idea: Each Query Message is forwarded intelligently based on what queries a peer answered in the past.

Components of ISM (for each node $u$)

a) Profile Mechanism, for each neighbor $N(u)$.

b) Peer Ranking Mechanism, for ranking peers locally and send a search query only to the ones that most likely will answer.

c) Similarity Function, for finding similar search queries.

d) Search Mechanism, for propagating queries based on local indexes.

Peer $q$
Intelligent Search Mechanism (ISM)

a) Profile mechanism.
- Maintains a list of past queries routed through that host.
- Every time a QueryHit is received the table is updated.

<table>
<thead>
<tr>
<th>Query</th>
<th>GUID</th>
<th>Connection</th>
<th>timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elections Bush Clinton</td>
<td>G439ID</td>
<td>Socket1</td>
<td>100002222</td>
</tr>
<tr>
<td>Super Bowl San Diego</td>
<td>F549QL</td>
<td>---</td>
<td>100065652</td>
</tr>
<tr>
<td>*****</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>Italy earthquake disaster</td>
<td>PN329D</td>
<td>Socket5</td>
<td>100022453</td>
</tr>
</tbody>
</table>

- The profile manager uses a Least Recently Used policy to keep most recent queries in repository.
- Profiles are kept for neighbors only so the cost for maintaining this cost is $O(Td)$, $T$ is a limiting factor per profile, $d$ is the degree of a node.
Intelligent Search Mechanism (ISM)

b) Peer Ranking Mechanism.

- Before forwarding a Query Message a peer performs an on-the-fly ranking of its peers to determine the best paths.
- We use the Aggregate Similarity of peer \( P_i \) to a query \( q \), computed by a peer \( P_k \) as:

\[
P_{sim}^{P_k}(P_i, q) = \sum_{q_j \text{ was answered by } P_i} Q_{sim}(q_j, q)^\alpha
\]
Intelligent Search Mechanism (ISM)

c) Similarity Function - The cosine.

- Assume that L is a set of all words (in Profile Manager)
  e.g. L = {elections, bush, clinton, super, bowl, san, diego, ... , italy, earthquake, disaster}

- We define an |L|-dimensional space where each query is a vector.
  If q = “italy disaster” => q (vector of q) = [0,0,0,...,1,0,1]

- Recall that we have a vector for each q_i stored in the Profile Manager (i.e. q_i)

\[
sim(q, q_i) = \cos(\tilde{q}, \tilde{q}_i) = \frac{\tilde{q} \cdot \tilde{q}_i}{\| \tilde{q} \|_2 \ast \| \tilde{q}_i \|_2}
\]
Intelligent Search Mechanism (ISM)

Search Mechanism

- Utilizes the Peer Ranking Mechanism to forward Queries to nodes that will potentially contain the info we are looking for.
Merging results

- multiple download sources
- partial downloads, reconnecting