Improving Performance in the Gnutella Protocol

Jonathan Hess          Benjamin Poon

University of California at Berkeley
Department of Computer Science
Cs294-4 Peer-to-Peer Systems
Outline

- Background
- Motivation
- Solution
  - Mirroring
  - Directed Search
- Results
- Possible Future Work
Background

- Gnutella
  - Protocol for distributed search
  - No centralization
  - Searches through query flooding

- Opponents
  - Censorship + threatening of Gnutella users
Motivation

1. Opponents cause ↓participation
2. ↓participation causes ↓replication of shared files
   - Same files being shared, but not as many copies
3. ↓replication causes
   - ↑workload for sharing peers
   - Need for deeper query depths
   - Overall decrease in performance
Solution

- Improve performance given decreased participation
  - Mirroring
  - Directed Search
Mirroring – Main Idea

- Achieve more replication by copying file to a willing peer (a mirror)
- Only replicate on demand
- Preserve blame on original sharer of file
  - i.e., mirrors should retain plausible deniability despite sharing the file
Mirroring Request Messages

- Mirror requestor (originator) sends Mirroring Request Message (MRM) to find a client to act as mirror
  - MRM(header, listeningPort, fileIndex)
- No need to flood
  - Clients pass MRM’s only on one randomly chosen outgoing connection
- $\text{MRM}_{\text{TTL}}$ should be relatively high
  - Prevents people from intercepting query traffic to see what file is
- Con: originator must stay in network in order for mirroring to occur
Mirroring – Sending MRMs

- Procedure per client sharing n files $F_1 \ldots F_n$
  1. Record demand $D_i$ (# uploads) for locally shared file $F_i$
  2. When $D_i > mirrorThresh_i$, request a mirror
     - Send MRM on one random outbound connection
  3. Having a new mirror means we shouldn’t create additional mirror as readily
     - $mirrorThresh_i += threshIncrement$
Mirroring – Receiving MRM s

1. Mirror M sends file transfer request for MRM.fileIndex to originator O
2. O receives request for fileIndex
3. O adds M to its list of mirrors of fileIndex
4. O sends M encrypted file associated with fileIndex
   - Preserves plausible deniability for mirror
   - Con: still a possibility for a client to figure out what original file was – how?
Mirroring – Using Mirrors

- **Procedure for originator of MRM s**
  - If originator has enough bandwidth
    - Serve files
  - If not enough bandwidth
    - Check if there are mirrors for file index
  - If no mirrors
    - Proceed according to original Gnutella protocol
  - If has mirrors
    - Multiplex requests over set of mirrors $M_1...M_x$
      - Send QueryHits as if they were from $M_i$ ($1 \leq i \leq x$) containing the decryption key
As the ratio of free-loaders to serving peers increases, search moves towards needle-in-a-haystack.

Flood excels at finding piles of hay.

Much research effort has gone into successive deepening and file indexing.

Directed search is not as well understood.
Directed Search – Main Idea

- Pay a one time up front cost for a bloom filter broadcast
- Nodes within N hops merge filter into a collection associated with each edge
  - Collection is depth aware
- Upon receiving a query, forward message to n edges with highest scores
Directed Search

- Query reaches $n^{query_{TTL}}$ nodes
- $n$ may be much smaller than out-degree and $query_{TTL}$ can be larger than normal TTLs
  - $n^{query_{TTL}} < \text{out-degree}^{TTL}$
- Reach more and better users
- Avoid free-loaders
Results

- Simulation: BloomNet
  - Models real-world Gnutella network as close as possible
    - Uses statistics from many previous measurement studies of Gnutella networks
- File sharing/requesting
  - Master filename list of 5072 files
  - Each client chooses to share certain number of files from master list
  - Queries generated by taking a random filename at most once from master list according to modified Zipf distribution (à la Efficient search in peer-to-peer networks, B. Yang, H. Garcia-Molina)
Results – Overview

Advantages

- BloomNet finds hits better than Gnutella
  - Uses approximately 3x less query bandwidth
  - As network size increases
    - Gap in performance increases
- BloomNet achieves higher % successful queries than Gnutella
  - Uses approximately 3x less query bandwidth

Disadvantages

- 20% more total bandwidth used to run BloomNet
  - Can be improved using different Bloom parameters
Results – Query Success

Query Success Over Bloom Parameters

Bloom Parameters (Depth/Buckets) vs. Query Success
Results – Query Bandwidth

Query Bandwidth Over Bloom Parameters

Bloom Parameters (Depth/Buckets) vs. Query Bandwidth
Results – Total Bandwidth

![Graph showing total bandwidth over Bloom parameters]

- Total Bandwidth Over Bloom Parameters
- X-axis: Bloom Parameters (Depth/Buckets)
- Y-axis: Total Bandwidth
- The graph illustrates the relationship between Bloom parameters and total bandwidth.
Possible Future Work

- Mirroring
  - More sophisticated demand realization techniques – gossiping protocols?

- Directed Search
  - Only highly-connected peers exchange Bloom Filters
  - Better score functions for edge selection
  - Better understanding of filter merging
Questions
<table>
<thead>
<tr>
<th>Basic</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>Router Type</td>
</tr>
<tr>
<td>1024</td>
<td>Number of Clients</td>
</tr>
<tr>
<td>200</td>
<td>Seconds To Run</td>
</tr>
<tr>
<td>2</td>
<td>Query Branching</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>TTLs</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Ping</td>
</tr>
<tr>
<td>5</td>
<td>Query</td>
</tr>
<tr>
<td>15</td>
<td>Mirror</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Bloom Filter</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>Buckets</td>
</tr>
<tr>
<td>4</td>
<td>Depth</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Max Desired Files</td>
</tr>
<tr>
<td>5</td>
<td>Mirroring Threshold</td>
</tr>
<tr>
<td>4</td>
<td>Client Out Degree</td>
</tr>
<tr>
<td>15</td>
<td>Chance Of Edge By Router</td>
</tr>
<tr>
<td>15</td>
<td>Seconds Per Connection</td>
</tr>
<tr>
<td>45</td>
<td>Seconds Per Ping</td>
</tr>
</tbody>
</table>

Simulate
Simulation Parameters

- Clients: 1024
- Bloom Depth: 3-4
- Bloom Size: 384-3072
- Ping TTL: 5
- Query TTL: 5-7
- Mirror TTL: 15