BitTorrent Protocol:
Priority Evaluation

Website:
http://www.sfu.ca/~csp6/

Simon Fraser University

Team #7
Charanpreet Parmar
FeiFan Jiang
Izaak Lee
Outline

• Introduction
• Motivation
• Related work and Reference Model
• BitTorrent Protocol
  – Seeds, Tracker, Peers
  – Tit for Tat
  – Super Seeding
• Implementation
• Conclusion
• Future Work
• BitTorrent is a Peer-to-Peer Network designed by Bram Cohen in April 2001
  – Allows users to connect directly to other users over the world wide web to share files.

• BitTorrent is responsible for over 18% of the traffic generated on the web.
  – More than 150 million people users
The question we want to analyze are:

- What are the weakness of the BitTorrent protocol?
- Can we improve on the efficiency of the network?
- Scalability?

In the end, we just want to get our files fast
Related Work and Reference Model

• Analysis of Live Video Streaming Over BitTorrent Peer-to-peer Protocol
  – By: Susan Herzarkhani and Milad Maleksabet

• BitTorrent in ns-2.29
  – By: Kolja Eger
  – It is not the entire BitTorrent Protocol
  – Predominately interested in the efficiency of the data transfer between peers
BitTorrent Protocol

- BitTorrent protocol has three parties:
  - Tracker: Is the list of the users sharing the same file, held at the server.
  - Users:
    - Seeder: One who has all the blocks of a specific file and sole purpose is to upload to peers.
    - Peer/leecher: A user who has none or some of the blocks of a file, but not the entire thing.
  - Server: provide torrent file that required joining a specific swarm

- P2P use Tit-for-Tat strategy
  - Equivalent retaliation
BitTorrent Protocol

- The parties combined are called the “swarm”

- Peers are responsible for sharing pieces it has with other peers

- BitTorrent downloads and uploads files in pieces
  - Ex. 1 file, 2000 pieces of data
• Random first piece: when a peer has no pieces, it will request for a random piece (not the rarest piece)

• Rarest First piece: find the minimal occurrence piece and set it to a high value

• Super Seeding: A seeder masquerades itself as peer, and as peers enter the swarm it will inform them it has a piece of data not yet in the swarm (Rarest). It will continue to do so until all the pieces needed to complete the file is in the swarm.
## BitTorrent Protocol

### Azureus BitTyrant

<table>
<thead>
<tr>
<th>Upload allocation</th>
<th>Up:Down</th>
<th>Rec:own</th>
<th>Client</th>
<th>T</th>
<th>Pieces</th>
<th>%</th>
<th>Last Tyrant Action</th>
<th>Do...speed</th>
<th>Up...speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>69.7 KB</td>
<td>n</td>
<td>1:1.18</td>
<td>1.01:1</td>
<td>Azureus 2.5.0.0</td>
<td>R</td>
<td></td>
<td>73.0% and their max upload rate.</td>
<td>11.2 kB/s</td>
<td>8.9 kB/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>0.64:1</td>
<td>1:∞</td>
<td>Azureus 2.5.0.0</td>
<td>L</td>
<td></td>
<td>51.9% of the last 60 seconds.</td>
<td>999 B/s</td>
<td>67 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>0.98:1</td>
<td>1:22:1</td>
<td>BitTorrent 1.1</td>
<td>R</td>
<td></td>
<td>21.7% choked.</td>
<td>0 B/s</td>
<td>0 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:59:1</td>
<td>1:13:1</td>
<td>Azureus 2.5.0.0</td>
<td>L</td>
<td></td>
<td>49.3% and their max upload rate.</td>
<td>5.7 kB/s</td>
<td>6.1 kB/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>Azureus 2.5.0.0</td>
<td>R</td>
<td></td>
<td>97.0% and their max upload rate.</td>
<td>1.4 kB/s</td>
<td>5.7 kB/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:34:1</td>
<td>1:∞</td>
<td>BitTorrent 0.81</td>
<td>L</td>
<td></td>
<td>100.0% Peer is a seed.</td>
<td>2.7 kB/s</td>
<td>0 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>Azureus 2.5.0.0</td>
<td>L</td>
<td></td>
<td>100.0% Peer is a seed.</td>
<td>1.2 kB/s</td>
<td>32 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>Unknown 0:0:0:0</td>
<td>L</td>
<td></td>
<td>46.1% of the last 60 seconds.</td>
<td>536 B/s</td>
<td>12 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>BitTorrent 0.62</td>
<td>L</td>
<td></td>
<td>100.0% Peer is a seed.</td>
<td>0 B/s</td>
<td>0 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>Azureus 2.5.0.0</td>
<td>L</td>
<td></td>
<td>100.0% Peer is a seed.</td>
<td>0 B/s</td>
<td>0 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>Azureus 2.5.0.0</td>
<td>L</td>
<td></td>
<td>100.0% Peer is a seed.</td>
<td>0 B/s</td>
<td>0 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>Azureus 2.2.0.2</td>
<td>L</td>
<td></td>
<td>100.0% Peer is a seed.</td>
<td>0 B/s</td>
<td>0 B/s</td>
</tr>
<tr>
<td>28.7 KB</td>
<td>n</td>
<td>1:∞</td>
<td>1:∞</td>
<td>BitTorrent 0.70</td>
<td>L</td>
<td></td>
<td>100.0% Peer is a seed.</td>
<td>210 B/s</td>
<td>0 B/s</td>
</tr>
</tbody>
</table>

### Piece Map

- **Azureus 2.5.0.0**: 51.9%
- **Unknown 0:0:0:0**: 48.1%

- Both have
- Peer has, you don't
- You have, peer doesn't
- Neither has
- Transferring
- Next Request
- Availability Count

- **Ratio**: NAT OK
- **1,186,745 Users**
- **IPs: 0 - 0/127**
- **85.0 kB/s**
- **(40k) 36.9 kB/s**
BitTorrent Protocol

- Peer communication
  - Operates over TCP
  - The peers operate in specific states:
    - Choked and un-choked
    - Interested and uninterested

- BitTorrent shares files through sharing priority
  - Those who share more get more
Download Priority in BitTorrent Protocol

• Currently:
  – Seeds look for the peers with the highest download rate only. Seeds will upload to those peers first.

• Our goal:
  – To analyze the effects of giving priority to peers who upload fastest priority.

  – Scalability
• We will simulate the standard BitTorrent protocol with ns2 and will serve as our baseline test
  – The network will consist of:
    • 1/3/1/5 Seed
    • 4/2/9/5 Peers
    • Upload speed of all peers 500 KB/s, except one which has 1000 KB/s
    • Download speed held constant 1000 KB/s
    • 1 MB file size
    • Super Seeding is enabled
  – What we are interested in is the time it takes for all Peers to have all the blocks to complete a file.
Tools Used:

- MATLAB – Parsing data and generating meaningful plots
- Ns-2.29 – Simulating various scenarios
- NAM – Add visualization of network activity
### Results

- **Simulation with 4 peers, 1 seed**

<table>
<thead>
<tr>
<th>Node</th>
<th>Start Time (s)</th>
<th>First Piece</th>
<th>Last Piece</th>
<th>Finish Time (s)</th>
<th>Time to Download (s)</th>
<th>Upload Rate (kbps)</th>
<th>Download Rate (KBps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>38.1814</td>
<td>N/A</td>
<td>500</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>12.5581</td>
<td>35.8674</td>
<td>38.1814</td>
<td>35.8674</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>5.37943</td>
<td>38.1814</td>
<td>38.1814</td>
<td>38.1814</td>
<td>500</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>7.74281</td>
<td>28.617</td>
<td>38.1814</td>
<td>28.617</td>
<td>500</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>16.5768</td>
<td>32.5465</td>
<td>38.1814</td>
<td>32.5465</td>
<td>500</td>
<td>1</td>
</tr>
</tbody>
</table>
Download Speed (Node 2)
Upload Speed (Node 2)
**Results**

- Simulation with 2 peers, 3 seeds

<table>
<thead>
<tr>
<th>Node</th>
<th>Start Time (s)</th>
<th>First Piece</th>
<th>Last Piece</th>
<th>Finish Time (s)</th>
<th>Time to Download (s)</th>
<th>Upload Rate (kbps)</th>
<th>Download Rate (MBps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>17.7353</td>
<td>N/A</td>
<td>500</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>17.7353</td>
<td>N/A</td>
<td>500</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>17.7353</td>
<td>N/A</td>
<td>500</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2.95244</td>
<td>13.238</td>
<td>17.7353</td>
<td><strong>13.238</strong></td>
<td><strong>1000</strong></td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>7.63118</td>
<td>17.7353</td>
<td>17.7353</td>
<td><strong>17.7353</strong></td>
<td>500</td>
<td>1</td>
</tr>
</tbody>
</table>
Download Speed (Node 4)
Upload Speed (Node 4)
### Results

- **Simulation with 9 Peers, 1 Seed**

<table>
<thead>
<tr>
<th>Node</th>
<th>Start Time (s)</th>
<th>First Piece</th>
<th>Last Piece</th>
<th>Finish Time (s)</th>
<th>Time to Download (s)</th>
<th>Upload Rate (kbps)</th>
<th>Download Rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>48.6981</td>
<td>N/A</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>15.8133</td>
<td>43.7755</td>
<td>48.6981</td>
<td>43.7755</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>8.08121</td>
<td>47.8112</td>
<td>48.6981</td>
<td>47.8112</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>12.6035</td>
<td>45.338</td>
<td>48.6981</td>
<td>45.338</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>30.6783</td>
<td>48.6981</td>
<td>48.6981</td>
<td>48.6981</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>17.3172</td>
<td>47.4061</td>
<td>48.6981</td>
<td>47.4061</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>22.1994</td>
<td>46.2685</td>
<td>48.6981</td>
<td>46.2685</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>31.5069</td>
<td>40.1077</td>
<td>48.6981</td>
<td>40.1077</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>27.9826</td>
<td>39.4187</td>
<td>48.6981</td>
<td>39.4187</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>27.2416</td>
<td>38.0289</td>
<td>48.6981</td>
<td>38.0289</td>
<td>500</td>
<td>1000</td>
</tr>
</tbody>
</table>
Download Speed (Node 6)
Upload Speed (Node 6)

Total Upload speed for node 6

Upload speed for node 6 per node
## Results

**Simulation with 5 Peers and 5 Seeds**

<table>
<thead>
<tr>
<th>Node</th>
<th>Start Time (s)</th>
<th>First Piece</th>
<th>Last Piece</th>
<th>Finish Time (s)</th>
<th>Time to Download (s)</th>
<th>Upload Rate (kbps)</th>
<th>Download Rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>22.085</td>
<td>-1</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>22.085</td>
<td>-1</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>22.085</td>
<td>-1</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>22.085</td>
<td>-1</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>22.085</td>
<td>-1</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>2.07539</td>
<td>9.36855</td>
<td>22.085</td>
<td>9.36855</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>8.69043</td>
<td>20.4494</td>
<td>22.085</td>
<td>20.4494</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>20.7728</td>
<td>22.085</td>
<td>22.085</td>
<td>22.085</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>10.6244</td>
<td>21.1587</td>
<td>22.085</td>
<td>21.1587</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>16.0654</td>
<td>19.0444</td>
<td>22.085</td>
<td>19.0444</td>
<td>500</td>
<td>1000</td>
</tr>
</tbody>
</table>
Download Speed (Node 6)
Upload Speed (Node 6)
Conclusion

• Peers with higher upload rates do not always perform better
  – Show inefficient use of resources
  – Leechers are hogging up resources
Future Work

• For us:
  – Look at the network with different priority implemented in the seeder
  – Determine the scalability and efficiency of the network.

• For more advance users:
  – Complete the BitTorrent for ns2.35
References


Questions?
Appendix - Tracker

- A Tracker manages the swarm
- Stores statistics about torrent
- Main role to find peers and start communication
- A HTTP/HTTPS service which works on port 6969
A torrent file is a bencoded dictionary with the following keys:

- **announce** - the URL of the tracker
- **info** - this maps to a dictionary whose keys are dependent on whether one or more files are being shared:
  - **name** - suggested file/directory name where the file(s) is/are to be saved
  - **piece length** - number of bytes per piece.
  - **pieces** - a hash list.
  - **length** - size of the file in bytes
  - **files** - a list of dictionaries each corresponding to a file (only when multiple files are being shared). Each dictionary has the following keys:
    - **path** - a list of strings corresponding to subdirectory names, the last of which is the actual file name
    - **length** - size of the file in bytes.
- All strings must be UTF-8 encoded.
Appendix – Hand Shake

- Handshaking is performed as follows:
  - The handshake starts with character 19 (base 10) followed by the string ‘BitTorrent Protocol’.
  - A 20 byte SHA1 hash of the bencoded info value from the metainfo is then sent. If this does not match between peers the connection is closed.
  - A 20 byte peer id is sent which is then used in tracker requests and included in peer requests. If the peer id does not match the one expected, the connection is closed.