Simon Fraser University

BitTorrent Protocol:
Priority Evaluation

Team #7
Charanpreet Parmar
FeiFan Jiang
Izaak Lee

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

ENSC 427 - Network Communication,
Spring 2012

Outline

- Introduction
- Motivation
- Related work and Reference Model
- BitTorrent Protocol
  - Seeds, Tracker, Peers
  - Tit for Tat
  - Super Seeding
- Implementation
- Conclusion
- Future Work

Introduction

- BitTorrent is a Peer-to-Peer Network designed by Bram Cohen in April 2001
  - Allows users to connect directly to other users over the wold wide web to share files.
- BitTorrent is responsible for over 18% of the traffic generated on the web.
  - More than 150 million people users

Motivation

- 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

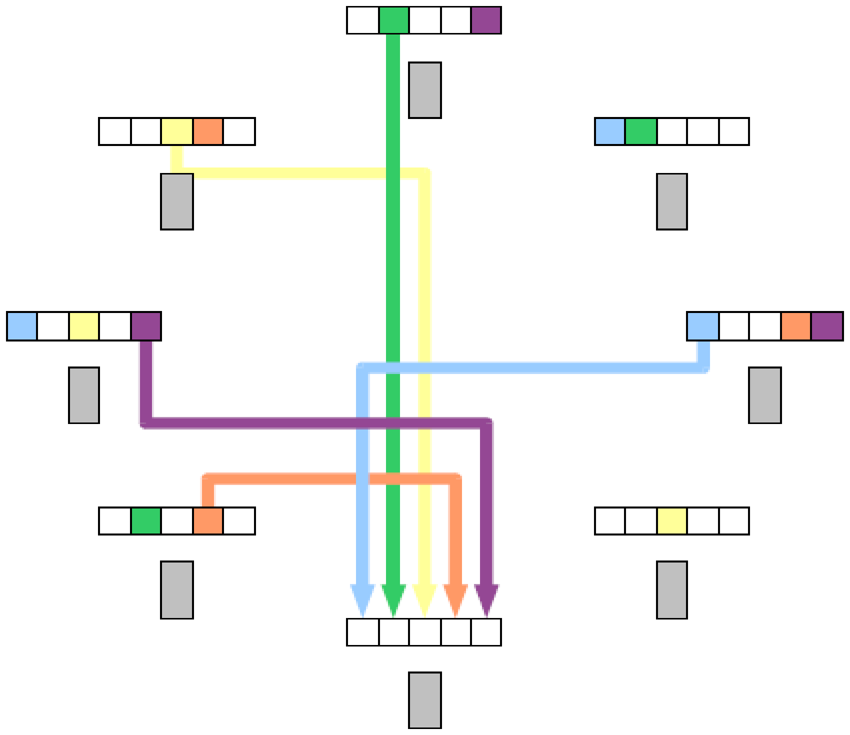
- BitTorrent in ns-2.29
  - By: Kolja Eger
  - It is not the entire BitTorrent Protocol
  - Predominately interested in the efficiency of the data transfer between peers

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

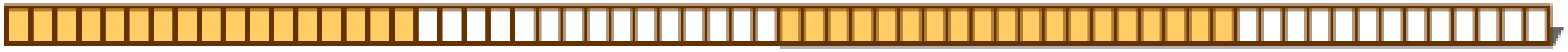
- The parties combined are called the "swarm"
- Peers are responsible for sharing pieces it has with other peers
- BitTorrent downloads and uploads files in pieces
  - Ex. 1 file, 2000 pieces of data



BitTorrent Protocol

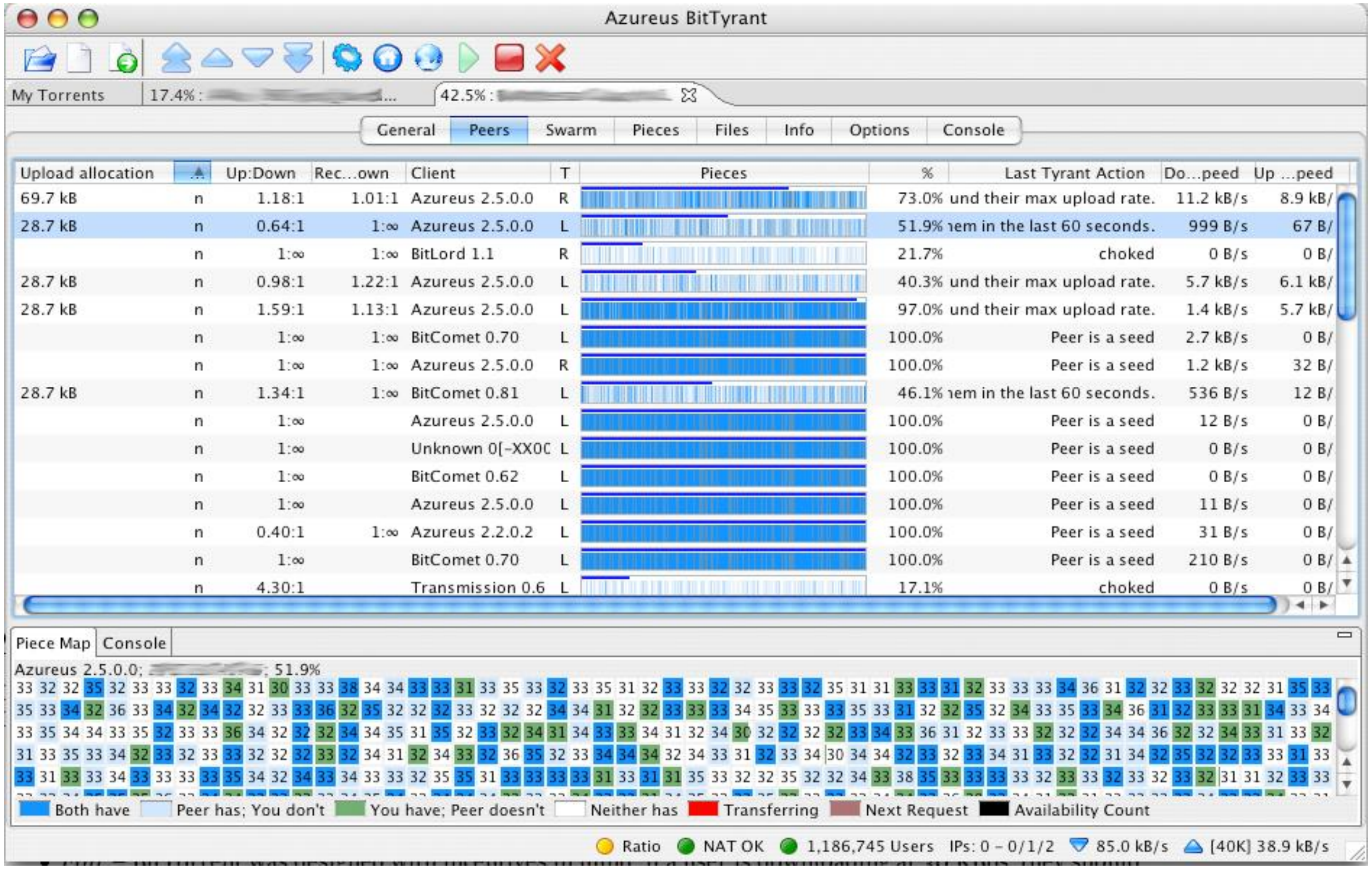
- Random first piece: when a peer has no pieces, it will request for a random piece (not the rarest piece)

- 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

My Torrents | 17.4% : [redacted]... | 42.5% : [redacted]

General | Peers | Swarm | Pieces | Files | Info | Options | Console

Upload allocation	▲	Up:Down	Rec...own	Client	T	Pieces	%	Last Tyrant Action	Do...peed	Up ...peed
69.7 kB	n	1.18:1	1.01:1	Azureus 2.5.0.0	R		73.0%	und their max upload rate.	11.2 kB/s	8.9 kB/
28.7 kB	n	0.64:1	1:∞	Azureus 2.5.0.0	L		51.9%	ıem in the last 60 seconds.	999 B/s	67 B/
	n	1:∞	1:∞	BitLord 1.1	R		21.7%	choked	0 B/s	0 B/
28.7 kB	n	0.98:1	1.22:1	Azureus 2.5.0.0	L		40.3%	und their max upload rate.	5.7 kB/s	6.1 kB/
28.7 kB	n	1.59:1	1.13:1	Azureus 2.5.0.0	L		97.0%	und their max upload rate.	1.4 kB/s	5.7 kB/
	n	1:∞	1:∞	BitComet 0.70	L		100.0%	Peer is a seed	2.7 kB/s	0 B/
	n	1:∞	1:∞	Azureus 2.5.0.0	R		100.0%	Peer is a seed	1.2 kB/s	32 B/
28.7 kB	n	1.34:1	1:∞	BitComet 0.81	L		46.1%	ıem in the last 60 seconds.	536 B/s	12 B/
	n	1:∞		Azureus 2.5.0.0	L		100.0%	Peer is a seed	12 B/s	0 B/
	n	1:∞		Unknown 0[-XX0C	L		100.0%	Peer is a seed	0 B/s	0 B/
	n	1:∞		BitComet 0.62	L		100.0%	Peer is a seed	0 B/s	0 B/
	n	1:∞		Azureus 2.5.0.0	L		100.0%	Peer is a seed	11 B/s	0 B/
	n	0.40:1	1:∞	Azureus 2.2.0.2	L		100.0%	Peer is a seed	31 B/s	0 B/
	n	1:∞		BitComet 0.70	L		100.0%	Peer is a seed	210 B/s	0 B/
	n	4.30:1		Transmission 0.6	L		17.1%	choked	0 B/s	0 B/

Piece Map | Console

Azureus 2.5.0.0; [redacted]; 51.9%

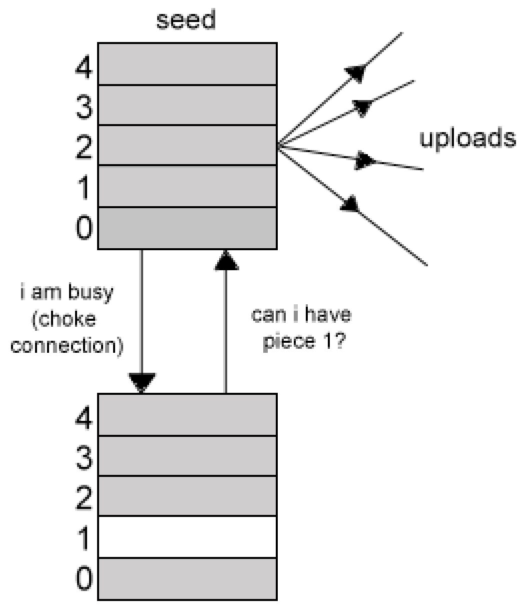
```
33 32 32 35 32 33 33 32 33 34 31 30 33 33 38 34 34 33 33 31 33 35 33 32 33 35 31 32 33 33 32 32 33 33 32 35 31 31 33 33 31 32 33 33 33 34 36 31 32 32 33 32 32 32 31 35 33
35 33 34 32 36 33 34 32 34 32 32 33 33 36 32 35 32 32 32 33 32 32 32 34 34 31 32 32 33 33 33 34 35 33 33 33 35 33 31 32 32 35 32 34 33 35 33 34 36 31 32 33 33 31 34 33 34
33 35 34 34 33 35 32 33 33 36 34 32 32 32 34 34 35 31 35 32 33 32 34 31 34 33 33 34 31 32 34 30 32 32 32 32 33 34 33 36 31 32 33 33 32 32 32 34 34 36 32 32 34 33 31 33 32
31 33 35 33 34 32 33 32 33 33 32 32 32 33 32 34 31 32 34 33 32 36 35 32 33 34 34 34 32 34 33 31 32 33 34 30 34 34 32 33 32 33 34 31 33 32 32 31 34 32 35 32 32 33 33 31 33
33 31 33 33 34 33 33 33 33 35 34 32 34 33 34 33 33 32 35 35 31 33 33 33 33 31 33 31 31 35 33 32 32 35 32 32 34 33 38 35 33 33 33 33 32 33 33 32 33 32 33 32 31 31 32 33 33
```

🟦	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 – 0/1/2 ▽ 85.0 kB/s △ [40K] 38.9 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

Implementation

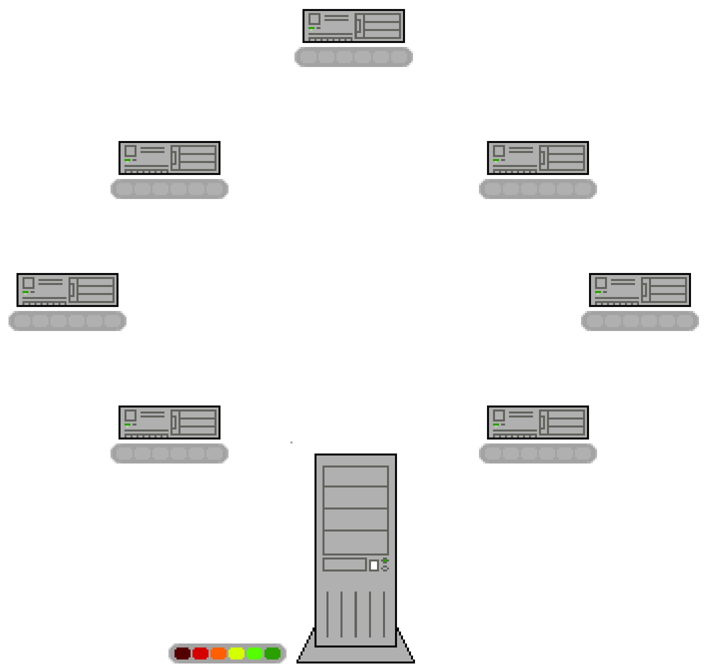
- We will simulate the standard BitTorrent protocol with ns2 and will serve as our baseline test
  - The network will consist of:
    - 1/3/1/5 Seed
    - 4/2/9/5 Peers
    - Upload speed of all peers 500 KB/s, except one which has 1000 KB/s
    - Download speed held constant 1000 KB/s
    - 1 MB file 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.

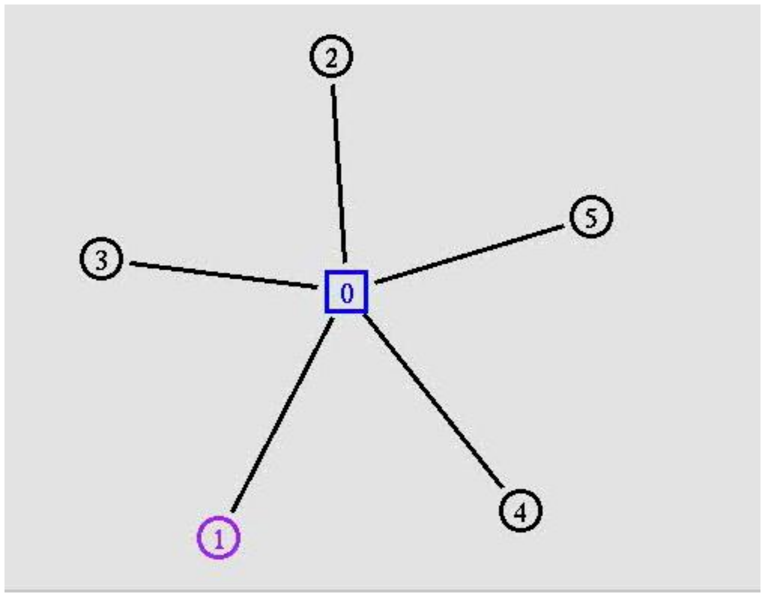
Simulation

Tools Used:

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

Simulation



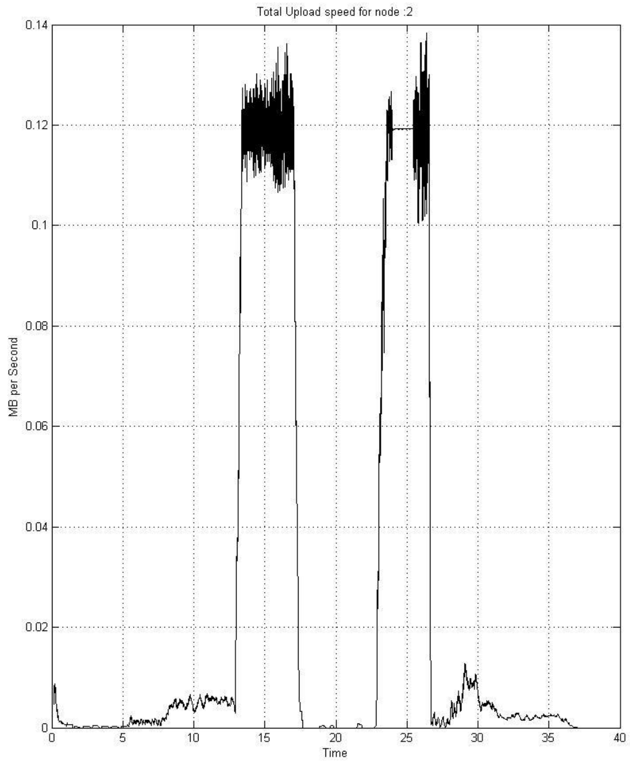


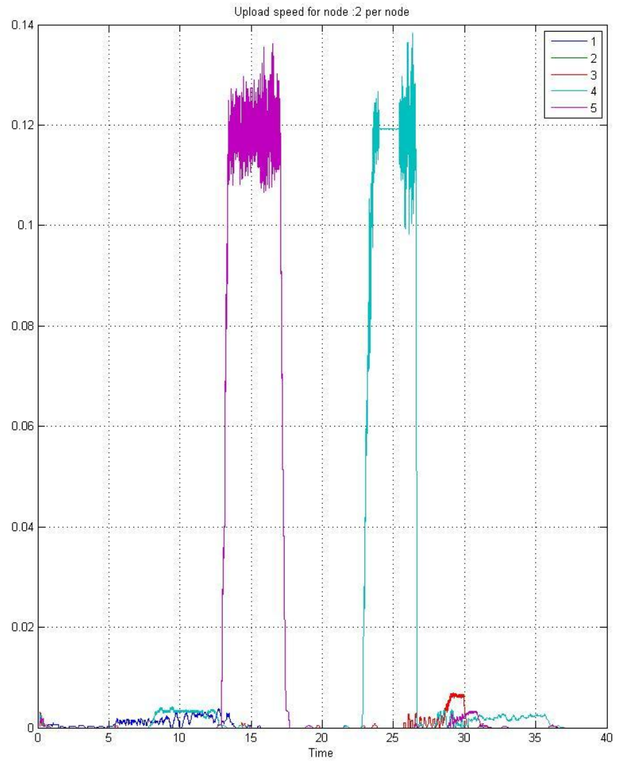
Results

- Simulation with 4 peers, 1 seed

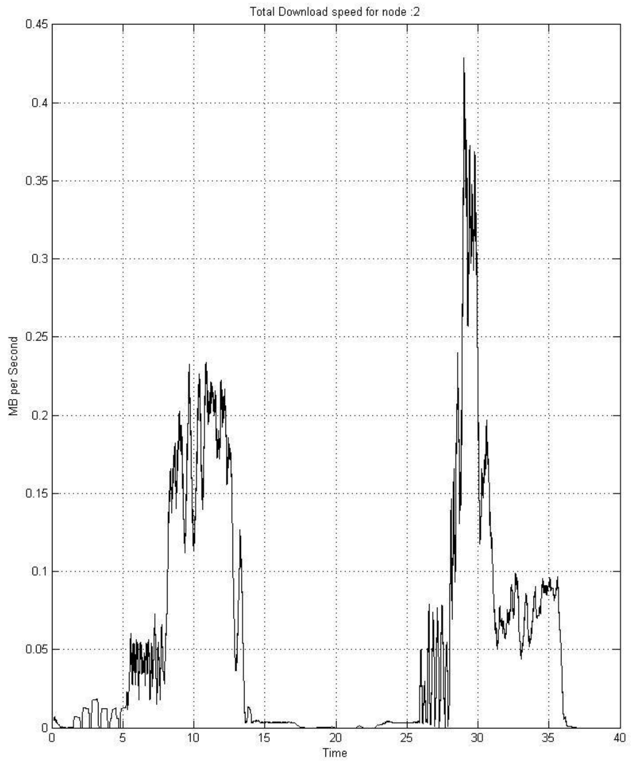
Node	Start Time (s)	First Piece	Last Piece	Finish Time (s)	Time to Download (s)	Upload Rate (kbps)	Download Rate (KBps)
1	0	0	-1	38.1814	**N/A**	**500**	1
2	0	12.5581	35.8674	38.1814	**35.8674**	**1000**	1
3	0	5.37943	38.1814	38.1814	**38.1814**	**500**	1
4	0	7.74281	28.617	38.1814	**28.617**	**500**	1
5	0	16.5768	32.5465	38.1814	**32.5465**	**500**	1

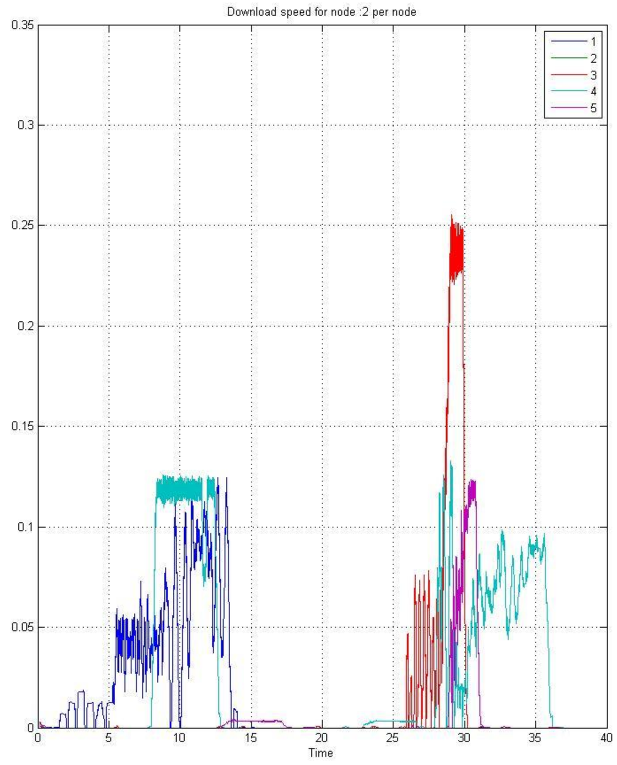
Download Speed (Node 2)





Upload Speed (Node 2)



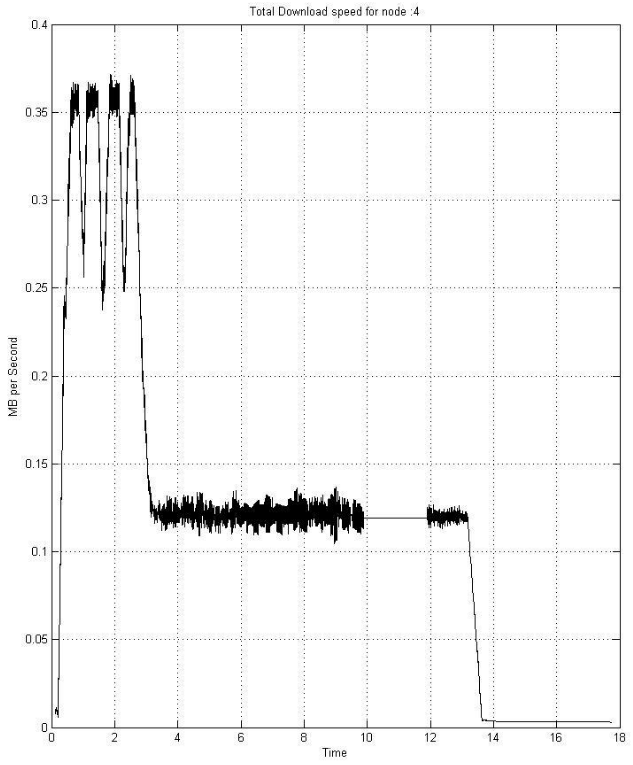


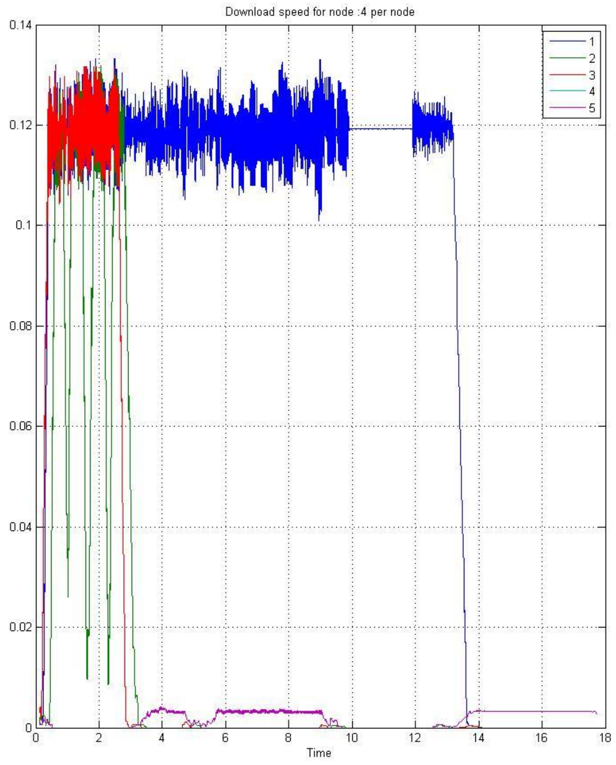
Results

- Simulation with 2 peers, 3 seeds

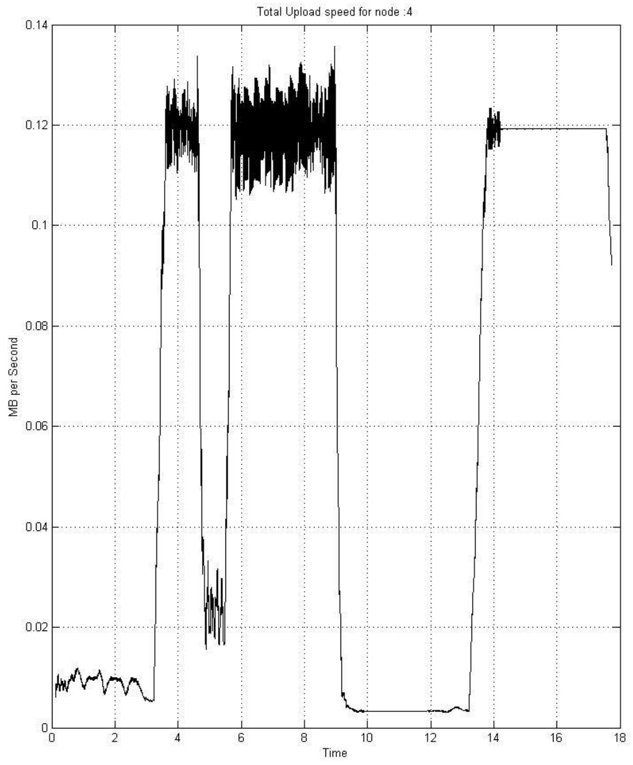
Node	Start Time (s)	First Piece	Last Piece	Finish Time (s)	Time to Download (s)	Upload Rate (kbps)	Download Rate (MBps)
1	0	0	-1	17.7353	**N/A**	**500**	1
2	0	0	-1	17.7353	**N/A**	**500**	1
3	0	0	-1	17.7353	**N/A**	**500**	1
4	0	2.95244	13.238	17.7353	**13.238**	**1000**	1
5	0	7.63118	17.7353	17.7353	**17.7353**	**500**	1

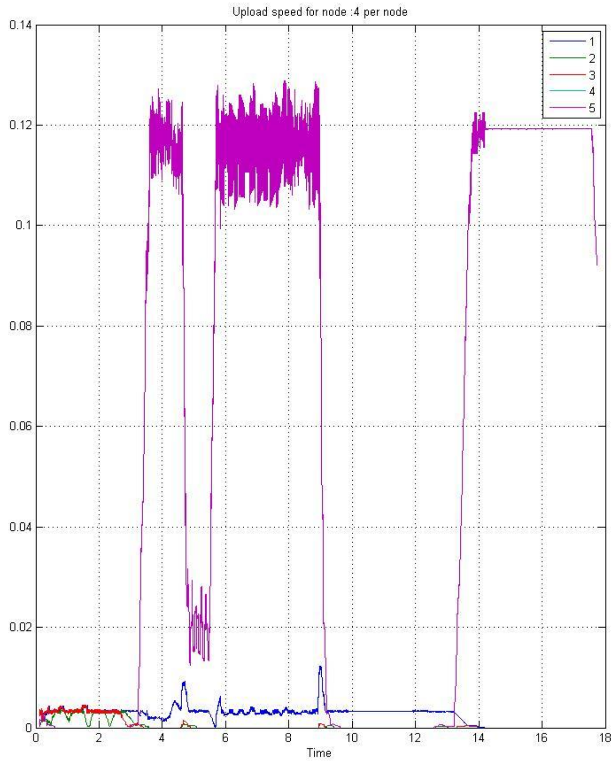
Download Speed (Node 4)





Upload Speed (Node 4)



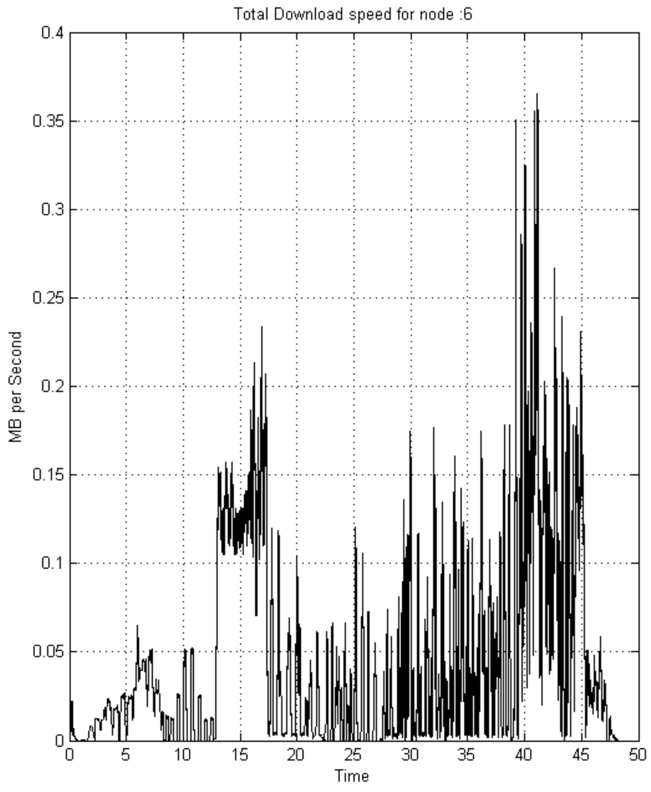


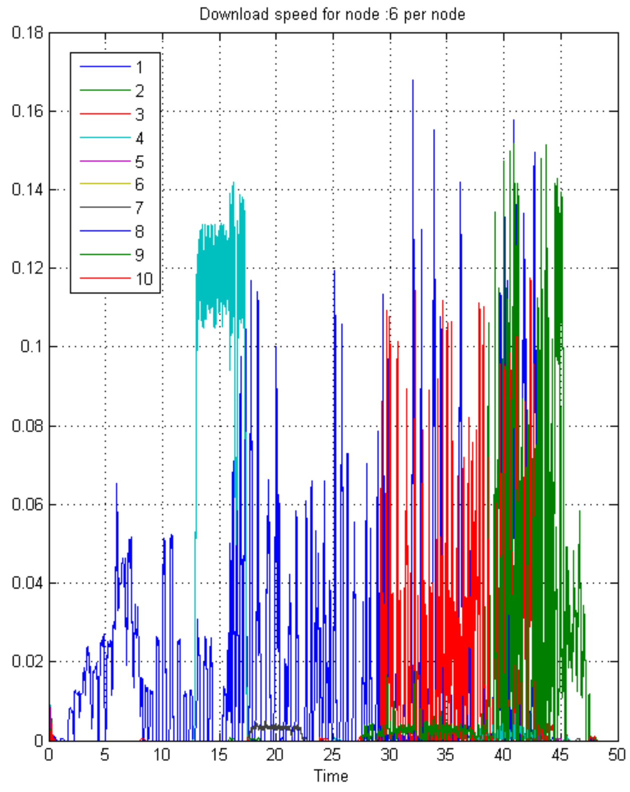
Results

- Simulation with 9 Peers, 1 Seed

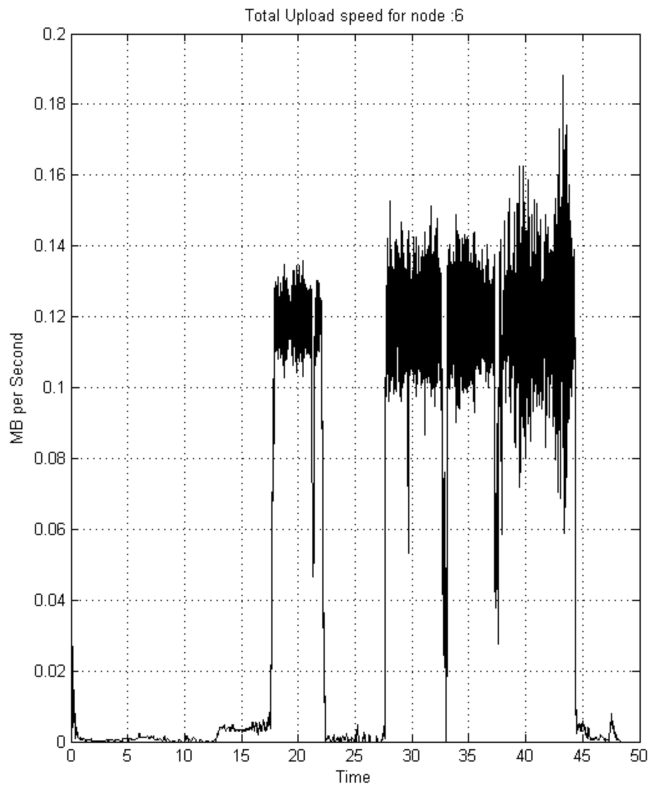
Node	Start Time (s)	First Piece	Last Piece	Finish Time (s)	Time to Download (s)	Upload Rate (kbps)	Download Rate (kbps)
1	0	0	-1	48.6981	**N/A**	500	1000
2	0	15.8133	43.7755	48.6981	**43.7755**	500	1000
3	0	8.08121	47.8112	48.6981	**47.8112**	500	1000
4	0	12.6035	45.338	48.6981	**45.338**	500	1000
5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

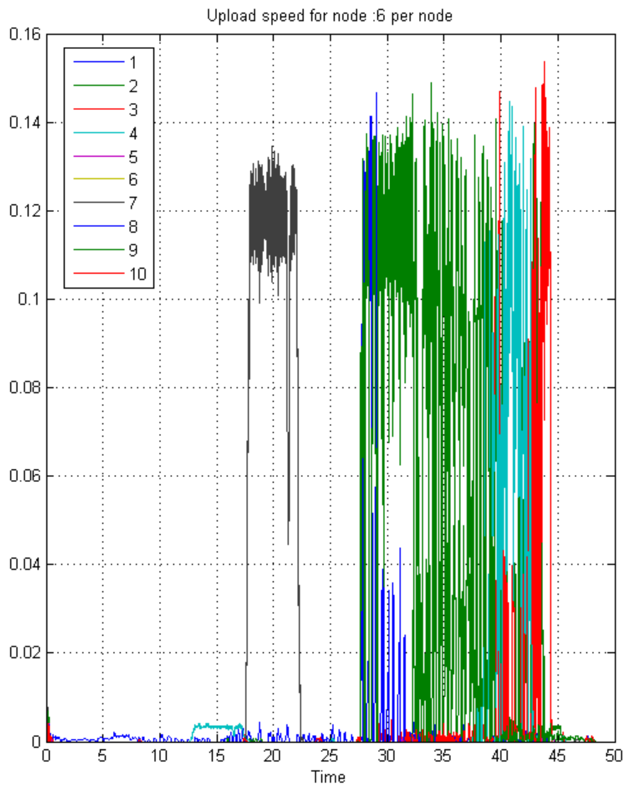
Download Speed (Node 6)





Upload Speed (Node 6)



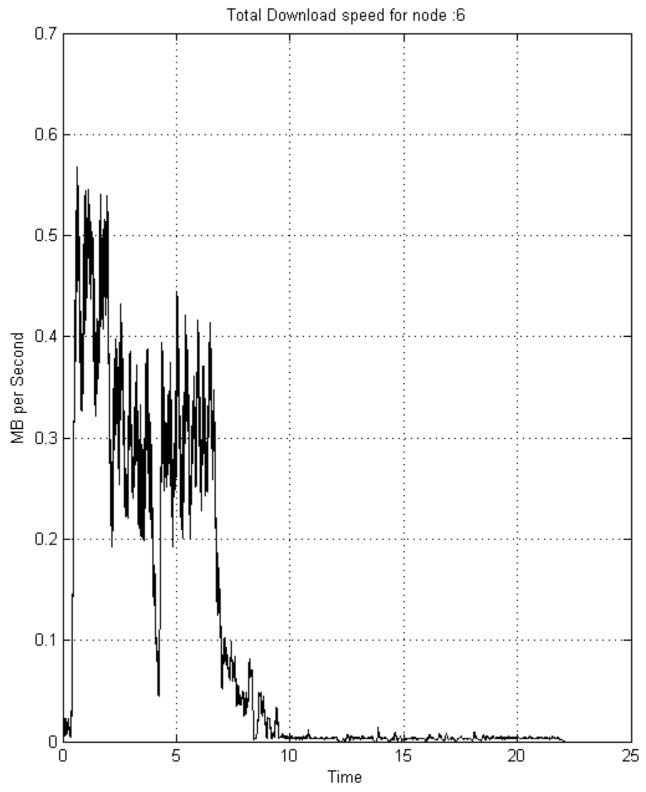


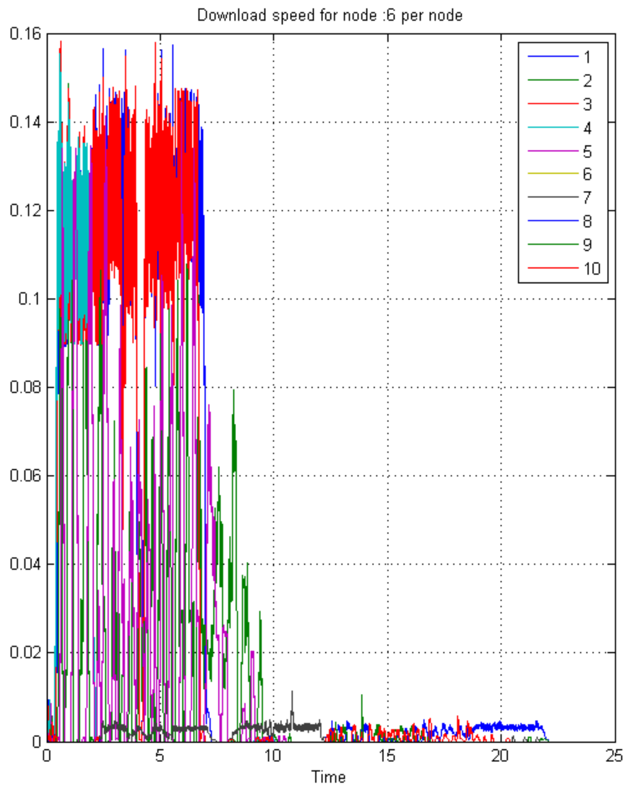
Results

- Simulation with 5 Peers and 5 Seeds

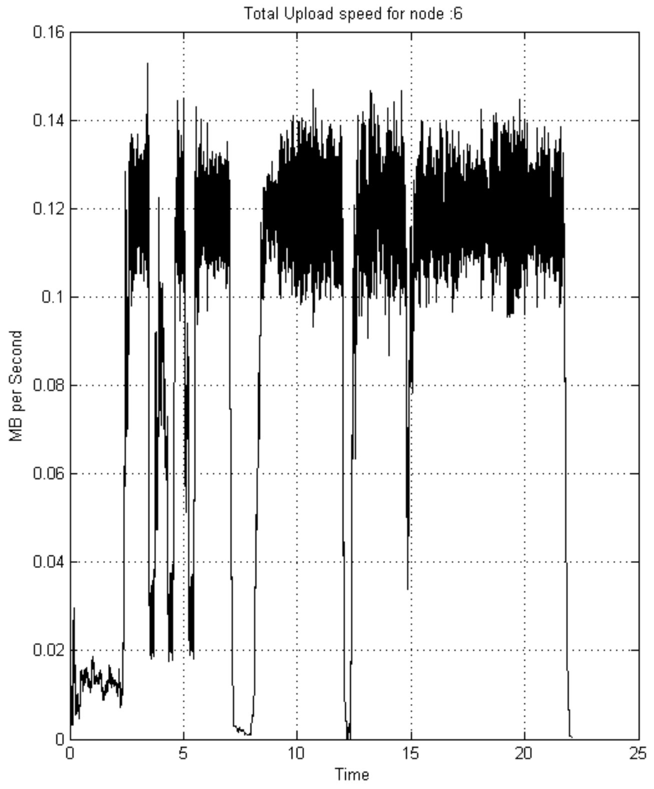
Node	Start Time (s)	First Piece	Last Piece	Finish Time (s)	Time to Download (s)	Upload Rate (kbps)	Download Rate (kbps)
1	0	0	-1	22.085	**-1**	**500**	1000
2	0	0	-1	22.085	**-1**	**500**	1000
3	0	0	-1	22.085	**-1**	**500**	1000
4	0	0	-1	22.085	**-1**	**500**	1000
5	0	0	-1	22.085	**-1**	**500**	1000
6	0	2.07539	9.36855	22.085	**9.36855**	**1000**	1000
7	0	8.69043	20.4494	22.085	**20.4494**	**500**	1000
8	0	20.7728	22.085	22.085	**22.085**	**500**	1000
9	0	10.6244	21.1587	22.085	**21.1587**	**500**	1000
10	0	16.0654	19.0444	22.085	**19.0444**	**500**	1000

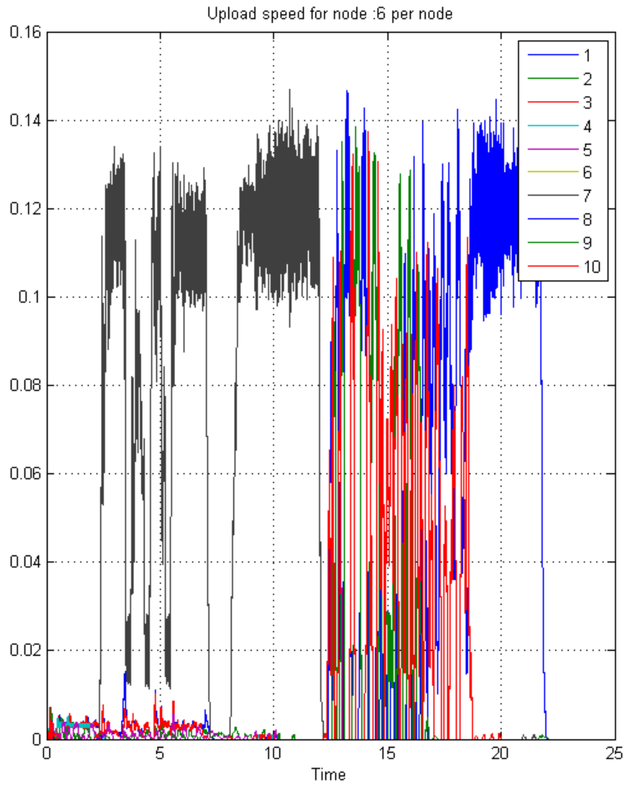
Download Speed (Node 6)





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

[1] B. Cohen, "BitTorrent," 10 1 2008. [Online]. Available: http://www.bittorrent.org/beps/bep_0003.html. [Accessed 1 3 2012].

[2] D. Schoder, K. Fischbach and C. Schmitt, "Core Concepts in peer to peer networking," University Cologne, Germany, 2005.

[3] K. Aberer, "Distributed Data Management Peer-to-Peer System," 2006.

[4] R. Steunmetz and K. Wehrle, "Peer-to-peer Systems," 2005.

[5] J. E. Berkes, "Descentralized Peer-to-Peer Network ArchitectureL Gnutella and Freenet," Winnipeg, 2003.

[6] J. Chung and M. Claypool, "NS By Example," Worcester polytechic institue, [Online]. Available: nile.wpi.edu/NS/. [Accessed 03 04 2012].

[7] M. M. Sasan Hezarkhani, "Analysis of Live Video Streaming Over Bittorrent Peer-to-Peer Protocol," Simon Fraser University, Vancouver, 2011.

[8] "Liberty Voice," 26 october 2010. [Online]. Available: http://www.libertyvoice.net/2010-10/bittorrent-still-dominates-global-internet-traffic/.

References

[9] A. Leon-garcia and I. Widjaja, Communication Networks: Fundamental Concepts and Key Architecture, New York: New York: Elizabeth A. Johns, 2004.

[10] D. Erman, D. Ilie and A. Popescu, "BitTorrent Session Characteristcs and Models".

[11] L. A., "Rarest First and Choke Algorithm are Enough".

[12] A. R. Bharambe and C. Herley, "Analyzing and Improving BitTorrent Perfomance".

[13] M. Baker and R. Lakhoo, "Peer-to-Peer Simulator".

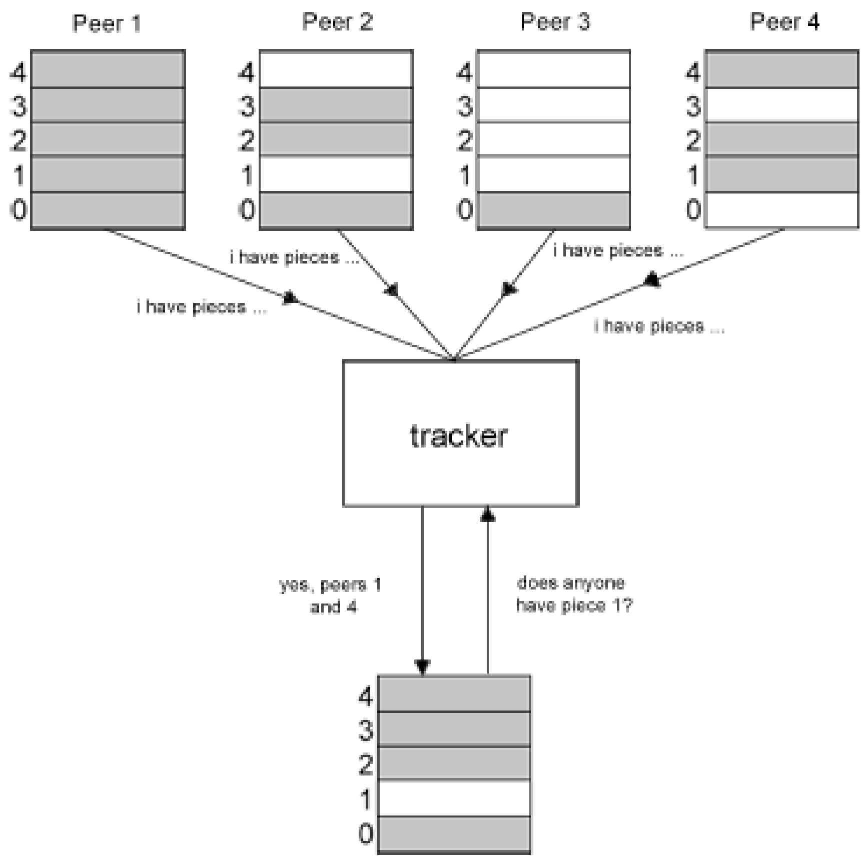
[14] E. Ayele, "Analysis and deployment of the BitTorrent protocol for community Ad-hol Network".

[15] K. Eger, "BitTorrent in ns-2," 11 January 2012. [Online]. Available: https://sites.google.com/site/koljaeger/bittorrent-simulation-in-ns-2. [Accessed 3 April 2012].

Questions?

Appendix - Tracker

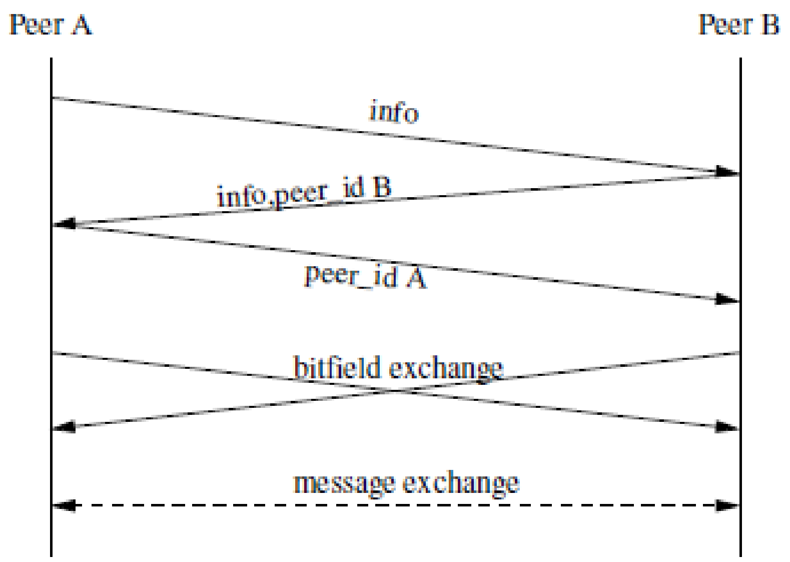
- A Tracker manages the swarm
- Stores statistics about torrent
- Main role to find peers and start communication
- A HTTP/HTTPS service which works on port 6969



Appendix – Torrent file

- 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.