CSC523: Analysis of the P2P BitTorrent Protocol

Abram Hindle

CSC523

University Of Victoria

abez@cs.uvic.ca

April 22, 2004

This Presentation

- What am I going to cover?
  - Introduction to BitTorrent
  - Explanation of BitTorrent Protocol
  - Literature Review
  - Block Distribution
  - Game Theory , Prisoners Dilemma and Tit For Tat
  - Byzantine Generals Problem
  - Hashing
  - Empirical Study
  - Summary

Introduction

- Terminology
  - P2P - Peer To Peer. Suggests multiple clients producing a mini-network.
  - Seeding - serving a file for download.
  - Leech - A client who is downloading from the seeders
  - Leeching - to download without contributing
  - Chunk - a piece of a file typically 64 KB to 256 KB in size.
  - Torrent - A file which provides a URL to the tracker as well contains a list of SHA1 hashes for the data being transfered. This is so that the hashes in the Torrent can be used to verify if the blocks received are valid or not.
  - Tracker - A middleman who informs the peers of all the other peers in the network.
  - Peer - A client to the network dedicated to a torrent.

- Seeder - A Peer who has all the blocks in a torrent.
- Choked - A connection is choked if not file data is passed through it. Control data may flow but the transmission of actual blocks will not.
- Interest - indicates whether a peer has blocks which other peers want.
- Snubbed - A peer acting poorly - not uploading - or sending bad control messages, usually disconnected or ignored.

Introduction

- What is BitTorrent [Coh01b]?
  - File Transfer Protocol
  - P2P
  - Leeches upload chunks to other leeches
  - Leeches relieve some of the load of distributing chunks from the seeders.

Introduction

- What isn’t BitTorrent?
  - Kazaa / Napster / Gnutella / etc..
  - Doesn’t provide a search mechanism
  - No chat, search, browsing, repositories
  - Essentially BitTorrent is just for downloading files

Introduction

- Why use BitTorrent.

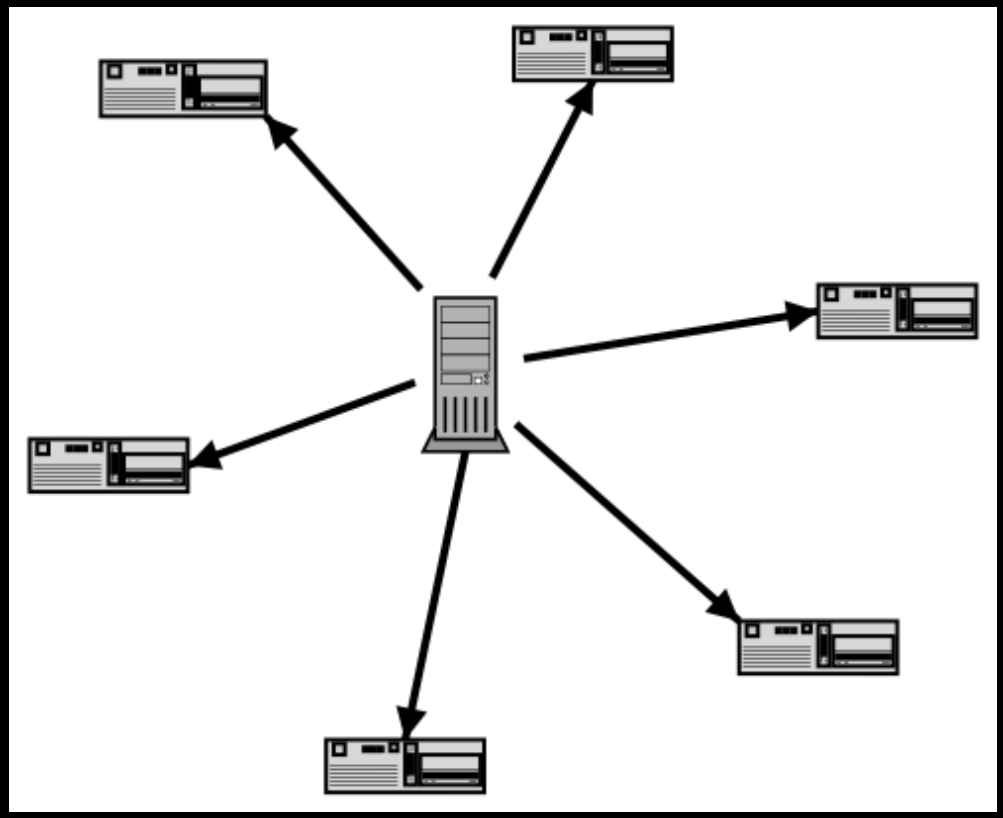

Figure 1: A Classical Distribution Scheme [Coh01a]

Introduction

- Why use BitTorrent.

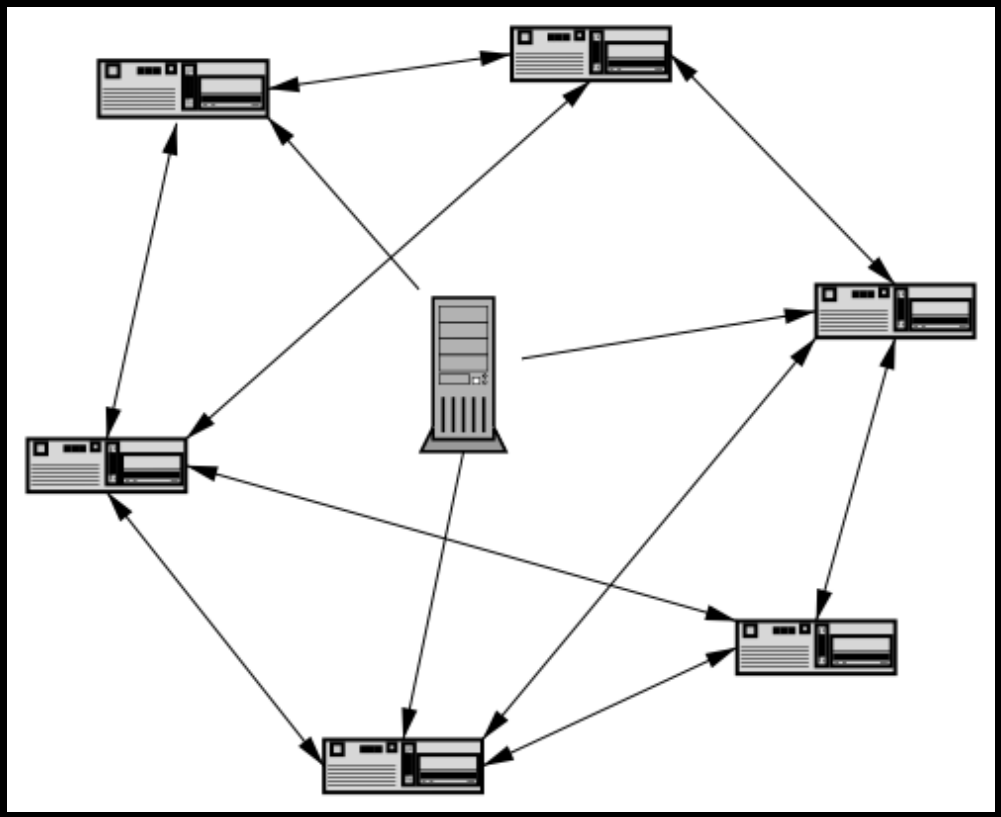

Figure 2: A BitTorrent Distribution Scheme [Coh01a]

Introduction

- Why use BitTorrent.
  - You want to share one group of files with people
  - You don’t want to spend n*filesize bytes in bandwidth
  - You only want to share your file with a small community (e.g. not Kazaa)
  - Cross platform
  - Open protocol, anyone can implement a client

Introduction

- Who uses BitTorrent?
  - iD Software, to distribute Wolfenstien Enemy Territory
  - Redhat, to distribute CD iso images.
  - Pirates, to distribute big files such as music, movies, ebooks, tv and games.
  - Music Bootleggers, to distribute high quality live concert recordings.

BitTorrent Usage

- A Usual BitTorrent Configuration:
  - Tracker - administers the torrent, tracks historical data
  - Torrent - the File which contains a link to the tracker and hashes of each chunk of the distribution.
  - Seeder - the initial client who provides the file that is torrented.
  - The leech finds the torrent, the torrent points to a tracker. Then the leech receives a list of peers from the tracker.

BitTorrent Algorithm

- Seeders
  - Seeders release the rarest blocks first in random order.
  - Random order maintains a uniform distribution of blocks among peers.
  - A client that is good at distributing blocks will get more attention from the seeder. [Azu]
  - Seeder offloads work to the peers.

BitTorrent Algorithm

- Leeches and Peers
  - Chunks of files are traded.
  - Chunks are verified by hashes in the torrent.
  - Tit For Tat Algorithm.
    - Those clients who upload the fastest to other clients get faster download rates from those clients. Some determination is to do with how much the client shares.
  - BitTorrent is for file distribution, not ensuring 1 to 1 upload to download ratios.
  - Assumption is about single clients not about groups of clients.

Literature Review

- “Analyzing peer-to-peer traffic across large networks” by Sen and Wang [SW02]
- “A Framework for the simulation of Agents with Emotions”, [BB01] by Bazzan and Bordini
- “Emotional Pathfinding”, [TDL04] by Donaldson, Park and Lin
- “A Framework for the simulation of Agents with Emotions”, [BB01]
- “Notions of reputation in multi-agents systems: a review”, by Mui, Mohtashemi, and Halberstadt [MMH02]
- “Towards a Pareto-optimal solution in general-sum games”, by Sen, Airiau and Mukherjee [SAM03]
- “The Byzantine Generals Problem” [LSP82] by Lamport, Shostak and Pease
- “Scalable Byzantine Agreement” [LS04] by Lewis and Saia

- “FARSITE: Federated, Available and Reliable Storage for an Incompletely Trusted Environment” , by many authors at Microsoft Research [ABC+02]

Block Distribution

- How are block initially and then later distributed.
  - Random Block First and Reasoning
  - Rarest Block First.
  - Results from Model
    - $E[X] = m + 2(m - m/n)ln(m - m/n) + c.$

Game Theory

- How is Game Theory Involved with BitTorrent?
  - Pareto Efficiency
  - Tit For Tat

Other Problems

- Other problems consisted of
  - Byzantine Generals Problem
  - Hashing (SHA1)

Empirical Study

- What did it consist of?
  - 5 Computers + Network
  - Software, Process Communication and synchronization
  - Modifying the BitTorrent client.
  - Test Bed scripts

Conclusions

- BitTorrent is very interesting as it's problem domain crosses many fields
  - Game Theory
    - Tit For Tat algorithm works well to promote fairness.
  - Computer Networks and Reliability
    - Block Distribution
  - Security
    - Byzantine Generals Problem

Future Work

- To do:
  - Investigate the effectiveness of multiple clients on one machine versus one client using the same amount of upstream bandwidth.
  - Investigate the effects of never admitting you have any blocks.
  - Investigate the effects of externally limiting upstream bandwidth.
  - Dedicate bandwidth to the experiment such that the total network bandwidth does not alter the results.

References

[ABC+02] A. Adya, W. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer. Farsite: Federated, available, and reliable storage for an incompletely trusted environment, 2002.

[Azu] Azureus. Azureus java bittorrent client details. *http://azureus.sourceforge.net/details.php*.

[BB01] Ana L. C. Bazzan and Rafael H. Bordini. A framework for the simulation of agents with emotions. In *Proceedings of the fifth international conference on Autonomous agents*, pages 292–299. ACM Press, 2001.

[Coh01a] Bram Cohen. Bittorrent: Introduction. 2001. *http://www.bitconjurer.org/BitTorrent/introduction.html*.

[Coh01b] Bram Cohen. The offical bittorrent homepage. 2001.

http://bitconjurer.org/BitTorrent/.

[Coh03] Bram Cohen. Incentives build robustness in bittorrent. May 2003.

[LS04] Clifford Scott Lewis and Jared Saia. Scalable byzantine agreement, 2004.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[MMH02] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. Notions of reputation in multi-agents systems: a review. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 280–287. ACM Press, 2002.

[MU03] Michael Mitzenmacher and Eli Upfal. *Probabilistic Analysis and Randomized Algorithms: A First Course*. Brown University, 2003.

[par04] Pareto efficiency. 2004. http://en.wikipedia.org/wiki/Pareto_efficiency.

[Pla03] David Player. Bittorrent chokepoint detection and response. 2003. *http://groups.yahoo.com/group/BitTorrent/message/3473*.

[SAM03] Sandip Sen, Stephane Airiau, and Rajatish Mukherjee. Towards a pareto-optimal solution in general-sum games. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 153–160. ACM Press, 2003.

[SW02] Subhabrata Sen and Jia Wang. Analyzing peer-to-peer traffic across large networks. In *Proceedings of the second ACM SIGCOMM Workshop on Internet measurment*, pages 137–150. ACM Press, 2002.

[TDL04] Andrew Park Toby Donaldson and I-Ling Lin. Emotional pathfinding, 2004.