Consistency and Collaboration

Victoria Michalska

Abstract

I'm making an algorithm that should communicate more consistently and evenly with specific types of agents that will hopefully shrink the standard deviation for download times for low-bandwidth agents in P2P networks. I'm implementing BitTorrent, FairTorrent, hopefully BitMate, and maybe my own version of some combination of these algorithms and then empirically analyzing the resulting data.

1 Introduction

This paper is an empirical study regarding the interactions and quality of interactions between different types of algorithms interacting in a single P2P environment. It began as an interest in shrinking the standard deviations of the necessary number of rounds for a certain type of player to finish downloading all of the necessary pieces of the file, but now has expanded into an interest in paying attention to the kinds of interactions occurring between different types of agents and the rationale behind them. By understanding these, I will attempt to create my own version of one of the implemented algorithms that will enable more consistent relationships between certain types of players that should minimize the standard deviation. When analysing such protocols, the dependent variable identified will be primarily the avg. download time.

The standard approach in the field of P2P file sharing, BitTorrent, follows a Tit-for-tat strategy, which is generally assumed to possess robustness.[1] There are also other alternatives, which optimize for other factors, by imposing additional checks and methodologies in order to account for freeloaders and other more malicious protocols. FairTorrent is a protocol "that accurately rewards peers in accordance with their contribution" by locally maintaining a list of those from whom that given peer had downloaded, and uploading to them accordingly, implementing a similar request protocol as BitTorrent but uploading to all possible request in order depending on to whom the agent owed the highest number of blocks.[3] Another possible implementation that focuses on having lower bandwidth players collaborate, in opposition to FairTorrent is BitMate, which focuses on agents finding and exchanging with agents with a similar bandwidth level to itself.[2] This sort of agent will also be implemented and analysed in relationship to the other protocols.

Real World Implementation Aiming to shrink the standard deviation for low-bandwidth agents appeals to agents in file-sharing networks because it means that they can download files of a fixed size in a more consistent amount of time; the inclusion of this sort of algorithm for an agent, however, may also result in poorer performance for other agents in specific instances and may result in poorer performance from an optimization standpoint, resulting in less active engagement from the agents in real-time networks.[1] This sort of algorithm also has other implementations in

[1]This algorithm will be further explained once this project is more complete.

other kinds of peer-to-peer networks, however. This includes energy distribution networks[4] and online lending markets[5], because uneven and unreliable distribution can deincentivize participants and prevent mutually beneficial exchange from occurring.

2 Background and Related Work

BitTorrent implements a tit-for-tat strategy: in the context of P2P networks, this means that such an agent will only upload to a peer that had uploaded to it in the most recent round. It is given four unchoke slots, three of which are allocated based off which peer sent over the highest number of blocks last round, with ties broken symmetrically.[1] This, however, results in bandwidth going unused often. FairTorrent, on the other hand, attempts to create a compromise between incentivizing peer uploads by rewarding high-participating peers and utilizing bandwidth efficiently by not explicitly choking peers who do not contribute.[3]

Bitmate specifically means to increase the number of communications between low-bandwidth agents. I'm still working out the details of how this sort of algorithm works.

3 Methodology and Contributions

Inspired by the work of Khan and Saif, I will be simulating a P2P environment featuring different sets of agents with different algorithms and then analysing the resulting interactions that occurred between different players, attempting to use that data to see whether or not manipulating those most consistent interactions can result in a smaller amount of variation when it comes to the time needed to complete a exchange process. I will be specifically looking at the number of blocks exchanged between different sets of peers.

Later, I plan on experimenting with creating specific groupings with different ranges of bandwidths for different kinds of agents and seeing how changing the kind of distribution will effect my results.

3.1 Data

This project will be collecting data regarding average bandwidth times and the number of blocks exchanged between certain sets of peers, and then representing that data visually using a heatmap where darker squares represent a higher number of blocks. This data will be extracted from a P2P file-sharing network simulator.

3.2 Implementation

The major parameters fed into the simulator are the following:

1. **Number of pieces:** The number of components in the file being downloaded.
2. **Blocks per piece:** The number of blocks needed to be downloaded to finish a whole piece.
3. **Minimum bandwidth:** The minimum number of blocks possible for an agent to send to another agent in a single round.
4. **Maximum bandwidth:** The maximum number of blocks possible for an agent to send to another agent in a single round.

5. **Iterations:** The number of times for the simulation to be run.

6. **Agent, Count:** The type of agent (with its corresponding algorithm) and the number of such an agent. This can take multiple different agents and different counts.

BitTorrent and FairTorrent having been implemented alongside a method for generating a visual aid describing the quantity of interactions between different agents, I could begin to gather data. So far, data for three different sets of agents has been collected, holding all other parameters fixed, including number of agents: one for exclusively BitTorrents, one for exclusively FairTorrents, and one for half BitTorrents and half FairTorrents.

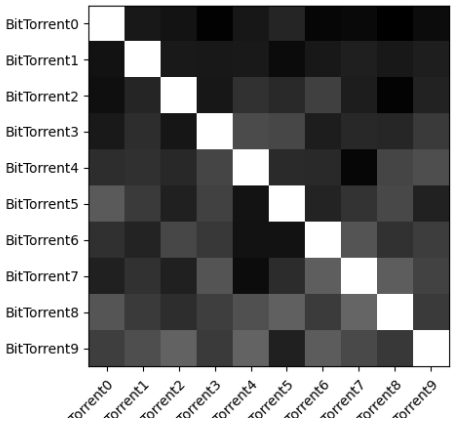

Figure 1: Seed,1 BitTorrent,10

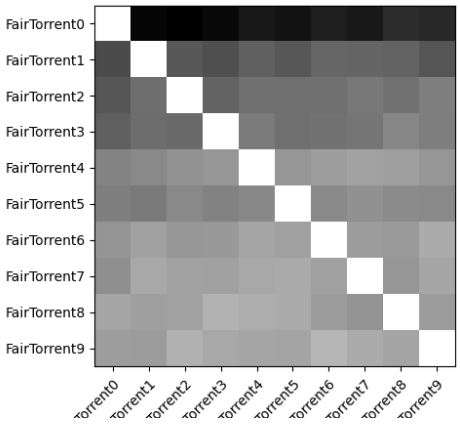

Figure 2: Seed,1 FairTorrent,10

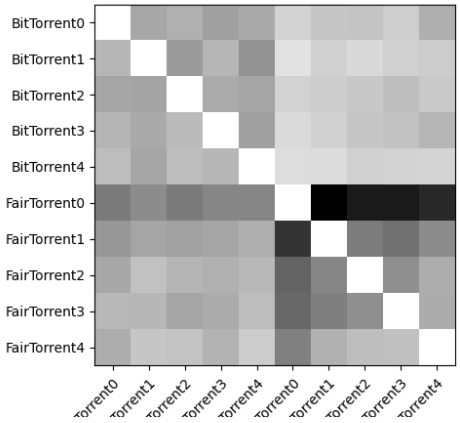

Figure 3: Seed,1 BitTorrent,5 FairTorrent,5

References

[1] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of PeertoPeer systems*, volume 6, 2003.

[2] Umair Waheed Khan and Umar Saif. Bittorrent for the less privileged. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, HotNets-X, New York, NY, USA, 2011. Association for Computing Machinery.

[3] Alex Sherman, Jason Nieh, and Clifford Stein. Fairtorrent: Bringing fairness to peer-to-peer systems. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, page 133–144, New York, NY, USA, 2009. Association for Computing Machinery.

[4] Esteban A. Soto, Lisa B. Bosman, Ebisa Wollega, and Walter D. Leon-Salas. Peer-to-peer energy trading: A review of the literature. *Applied Energy*, 283:116268, 2021.

[5] Hongchang Wang and Eric Overby. How does algorithmic trading influence investor participation in peer-to-peer online lending markets? In *53rd Hawaii International Conference on System Sciences, HICSS 2020, Maui, Hawaii, USA, January 7-10, 2020*, pages 1–10. ScholarSpace, 2020.