BitTorrent Fairness: Analysis and Improvements

Richard Thommes and Mark Coates

Department of Electrical and Computer Engineering
McGill University
3480 University St
Montreal, QC, Canada H3A 2A7
Email:{rthomm,coates}@tsp.ece.mcgill.ca

***Abstract*— BitTorrent is an extremely effective and popular peer-to-peer file distribution application. It differs from traditional peer-to-peer file-sharing applications in that large files are decomposed into blocks, and in order to download a file, a peer concurrently retrieves blocks from multiple peers. Measurement and simulation studies have suggested that although BitTorrent achieves excellent utilization of upload capacity, its *fairness* properties are less impressive. In this paper, we seek to understand, primarily through simulation analysis, the fairness properties of the exchange mechanism that lies at the core of the BitTorrent protocol. We focus on a specific fairness metric, defined as the ratio of bytes uploaded to that downloaded by each individual peer. We propose three modifications to the protocol, and examine their impact on the fairness peers experience.**

I. Introduction

BitTorrent is an extremely popular peer-to-peer application for sharing large files, based on the principle of decomposing a file into multiple small blocks. A peer can then download different blocks concurrently from multiple peers, and during the downloading process provide other peers with the blocks it has already retrieved. BitTorrent employs a rate-based "tit-for-tat" policy, whereby a peer chooses to upload to a small set of neighbouring peers which are providing it with the best download rates. This mechanism is intended to discourage free-riding (downloading without uploading) and promote fairness.

Measurement studies have indicated that BitTorrent displays excellent scalability and achieves high utilization of the available upload capacity of the network [1], [2]. These same studies and detailed simulation studies [3] have, however, called into question the fairness properties of the BitTorrent protocol. It has been observed that peers with high upload bandwidth frequently upload much more data than they download, with the opposite being the case for peers with low upload bandwidths. In this paper, we focus on the peer selection and search (*unchoking*) techniques that underpin the tit-for-tat policy, and conduct analysis and simulations of a simplified BitTorrent model to explore the fairness properties of these mechanisms. Our results illustrate that these techniques can induce substantial unfairness in "vanilla" BitTorrent, but the unfairness we observe is not as dramatic as that reported elsewhere [2], [3], indicating that other aspects of the protocol that we do not incorporate in our model exacerbate the unfairness. We propose several modifications to BitTorrent to improve the fairness and examine their impact through simulation.

The paper is structured as follows. In the remainder of the introduction, we provide an overview of the BitTorrent protocol, and discuss the relationship between our research and prior work. Section II describes the simplified and abstracted BitTorrent model that we analyse and simulate. In Section III we identify an equilibrium state for the system when the optimistic unchoke procedure is idealized, and demonstrate that this state provides a form of fairness. Section IV describes modifications we propose to enhance fairness and Section V analyses the simulation results. Section VI provides concluding remarks and indicates avenues of future research.

A. The BitTorrent Protocol

BitTorrent is a peer-to-peer application that aims to enable the fast and efficient distribution of large files [4]. Here we provide a brief overview; see [2]–[5] for more detailed descriptions. The primary difference between BitTorrent and other file-sharing applications operating on peer-to-peer networks such as e-Donkey [6] and Gnutella [7] is that the files are split into equal-sized *blocks* and peers download these blocks concurrently from multiple peers. For each *torrent* (file) available for download, there is a centralized *tracker* that keeps track of the peers currently in the system. When a peer wishes to download the torrent, it notifies the tracker, and receives a list containing a random subset of the other peers. The peer attempts to establish connections to these other peers, which become its neighbours upon success. The group of neighbours is called the *peerset* of a peer, and in practice numbers about 40.

Peers in the system are either *seeds* or *leechers*. Seeds have a complete copy of the file and are remaining in the system to provide blocks to others. Leechers are in the process of downloading the file, and can only upload the blocks they have already retrieved. Each peer strives to download blocks from other peers. Initially, when a peer needs to quickly acquire blocks to exchange, it accepts whatever blocks are made available, but later it chooses the blocks that are rarest amongst its neighbours in a *local rarest first* policy.

BitTorrent attempts to induce fairness and guard against free-riding through a rate-based tit-for-tat policy. Each peer maintains a small, constant number of concurrent uploads (usually 5), preserving the balance through a process called *choking*. At any moment a peer has a set of 5 unchoked

neighbours (those to which it is uploading) and a set of choked neighbours. Every ten seconds the peer evaluates the download rates it is receiving from its neighbours. If the lowest download rate provided by an unchoked neighbour is less than the highest provided by a choked neighbour, then the peer chokes the former and unchokes the latter. This peer selection policy attempts to establish the "fair" scenario where peers upload to and download from peers with similar bandwidths.

In addition to this peer selection policy, BitTorrent incorporates *optimistic unchoking*. Every thirty seconds a peer randomly chooses a neighbour and uploads to it. This is both a search procedure, allowing peers to discover neighbours with better upload capability, and also serves to bootstrap peers that have just joined by providing them with an initial set of blocks to exchange.

B. Relationship to Prior Work

Bharambe et al. report on simulation analysis of the BitTorrent protocol in [3], [8]. They examine fairness, concluding that the BitTorrent rate-based tit-for-tat policy fails to prevent unfairness (with some peers uploading up to six times as much as they download). In contrast to their simulator, which strives for a realistic simulation of almost all aspects of the BitTorrent protocol, we aim to assess in this paper the fairness properties of the *core* peer selection and search procedures underpinning BitTorrent. Hence we eliminate from our model and simulations many of the implementation issues such as initial seeding techniques, block selection, peerset restrictions, and endgame policies, some of which can serve to exacerbate unfairness.

Qiu et al. [9] proposed a fluid model for BitTorrent and analyzed the effectiveness of the tit-for-tat mechanism. They explored the behaviour of the system when peers can adjust the upload bandwidth devoted to BitTorrent exchanges with the goal of minimizing their upload rate (whilst maintaining maximum available download rate). They prove that there is a Nash equilibrium with this strategy in effect only if the network consists of groups of peers with the same maximum upload capacity. The equilibrium point occurs when peers set their upload bandwidth to its maximum. We assume in this work that peers choose to operate at this point; in Section III, we identify an equilibrium point that exists if the unchoking mechanism is idealized and discuss its fairness; this result is strongly related to Lemmas 1 and 2 and Proposition 1 of [9].

Bharambe et al. [3] also propose two modifications for addressing the unfairness: quick bandwidth estimation and block-level tit-for-tat. The first modification involves a rapid estimation of the upload capabilities of the peers in the peerset through some form of probing scheme. A peer can then avoid unchoking peers with much lower transfer capability. The modification is somewhat idealistic, because reliable bandwidth estimation is far from a trivial exercise. The block-level tit-for-tat approach enforces fairness, but can result in a reduction in upload capacity utilization because peers can potentially cease to upload whilst waiting to receive reciprocal blocks. We do not compare our proposals to these modifications here because the simplifications we adopt in order to focus on peer selection and optimistic unchoking do not permit a fair comparison.

II. Model Description

In this section we describe the details of the BitTorrent model used in our analysis and simulations. We make a number of simplifying assumptions:

- Every peer is always able to provide any other peer a desired block of the file.
- In all cases upload capacities rather than download capacities are the bottlenecks in data transfers.
- For each peer, its peerset – the set of other peers it is aware of and able to connect to – includes all the peers in the system.

These three assumptions imply that at any point in time a peer *i* is able to download from any other peer *j* if *j* wishes to upload to *i*. Limitations on download rates, restricted peersets and uneven block availability serve to reduce the number of possible connections that may exist between peers and hence interfere with the (un-)choking procedure. Since we are interested in assessing the inherent fairness of BitTorrent protocol peer selection and (un-)choking, we do not model these constraints.

In addition, we idealize network behaviour, assuming that:

- A peer always utilizes its full upload capacity, and is always sending data to five other peers. The upload rate to each of these peers is exactly one-fifth of upload capacity.
- Peers are able to measure download rates with perfect accuracy.
- Peers always send at full rate, i.e., the ramp-up time of a connection is negligible.

A. Simulator Description

We implemented the above BitTorrent abstraction as a discrete-time simulator in Matlab. Each of the N simulated peers has a fixed upload rate, normalized to fall in the interval $(0.05, 1]$ (this might correspond to the range of 50kbps to 1Mbps). Initially, upload rates are randomly chosen according to a uniform distribution and each peer randomly chooses the 5 peers to which it uploads. The peer selection (choking and unchoking) procedure occurs as in the BitTorrent protocol described in Section I-A, with peers calculating download rates every 10 seconds. Optimistic unchoking of a random peer is performed every 30 seconds. The simulation proceeds in 1-second time steps, with each peer's initial unchoke uniformly distributed between 0 and 9 seconds from the beginning of the simulation.

III. Theoretical Analysis

In this section we identify an equilibrium state for a system operating according to the model specified above, except that optimistic unchoking is replaced by an idealized mechanism where peers exchange truthful information about upload capabilities and establish a connection if both peers agree.

Consider N peers downloading from one another. Every peer has a fixed upload capacity, and no two peers have exactly the same capacity. Every peer acts in a greedy manner to maximize its download rate. A peer can upload to five different peers at one time, sending data to each at one fifth of its upload capacity. All connections are bidirectional: peer *i* uploads to peer *j* if and only if peer *j* uploads to peer *i*. To initiate a new connection, peer *i* sends out a request to peer *j* specifying the upload rate it can provide. Peer *j* responds with its offered upload rate. A connection is established only if both peers agree. At any point, either peer may close the connection.

The following lemma identifies the equilibrium state and is useful for quantifying the performance of our suggested improvements to the BitTorrent protocol, and we will utilize it in Section V. The corollary follows directly from the proof of the lemma.

Lemma 1: The system outlined above achieves an equilibrium point where peers form into $\lfloor \frac{N}{6} \rfloor$ disjoint groups of six and one group comprising the remaining peers. Group members upload to and download from each of the other members. Once this unique set of groups is established, no pair of peers will agree to form a new connection. Each peer is downloading at its maximum rate according to the system rules.

Corollary 1: The equilibrium point achieves a form of fairness: the download rate of a peer i cannot be increased without decreasing the download rate of a peer j with higher upload bandwidth.

Proof: Order the peers from 1 to N according in ascending order of their upload capacity. The peer with the highest upload capacity will henceforth be referred to as peer N, the one with the second highest upload capacity as peer $N-1$ etc. Consider peer N. The highest download rate it is able to achieve is if it is downloading from peers $N-1$, $N-2$,..., $N-5$. Thus, if peer N establishes a connection with each one of these peers, it will not agree to any subsequent connection requests. Since N offers the highest upload rate in the entire network, none of the five peers connected to it will drop the connection due to any new requests. Next, peer $N-1$ will achieve the highest download rate if it is downloading from N, $N-2$,.., $N-5$, and thus it will not agree to any new connections once it has these five established. This argument continues up to and including peer $N-5$.

Now consider peer $N-6$: if the first group is formed, it is unable to "convince" any of the five higher ranked peers to form a connection. This means the highest download rate it can achieve is if it establishes a connection with the four peers below it in ranking: $N-7, ..., N-11$. These peers, in turn, are also unable to join the first group and thus maximize their download rate if they form connections among each other. This argument can be continued inductively to all other peers, except peers $1, 2, .., K$, where $K = N \mod 6$. These K lowest ranked peers must form a group among each other, and each will only have $K-1$ outgoing/incoming connections. ■

IV. Proposed BitTorrent Modifications

In this section we propose three approaches for improving BitTorrent fairness. We treat each modification separately, as they cannot be combined. We define the *Instantaneous Fairness Ratio* (IFR) for an individual peer as the ratio of data uploaded to data downloaded during the last 10 seconds. Therefore, an IFR of less than 1 indicates a peer is downloading an excessive amount (relative to perfect fairness), and an IFR of greater than 1 indicates a peer is downloading an insufficient amount.

A. Conditional Optimistic Unchoke

The Conditional Optimistic Unchoke modification represents a minor change to the BitTorrent protocol. A peer performs an optimistic unchoke only if its IFR is greater than 1. Essentially, peers operate in a more cautious manner: if a peer has an IFR of less than 1, it is already downloading more than its fair share of data. Choking an outgoing connection is likely to change the set of peers from which it is downloading, and hence the peer risks eliminating or reducing its download surplus. Peers do not take this risk, thereby also forgoing some opportunities to potentially further reduce their IFR.

B. Multiple Connection Chokes

The Multiple Connection Chokes modification allows peers to choke/unchoke multiple connections each round. A peer calculates the *Connection Fairness* for each of the five peers to which it is uploading. This is simply the ratio of the peer's upload rate to a specific peer to the download rate from that peer. If the other peer is not sending any data, the connection fairness is defined as infinity. There are two parameters in the modification: the Threshold Ratio, which is the largest value a Connection Fairness can assume before the corresponding upload may be choked, and the Maximum Chokes (MC), which is the largest number of uploads a peer can choke per round. It initially appears tempting to set the Threshold Ratio to 1. However, unless two peers have exactly the same upload capacity, one will always face a Connection Fairness of less than 1. Thus, if the Threshold Ratio is not greater than 1, few connections persist. If during a given round the number of Connection Fairness values exceeding the Threshold Ratio is less than or equal to MC, the peer will choke all the unfair connections. Otherwise, it chokes only MC of the peers, chosen at random. For every choked connection, the peer considers the set of other peers currently uploading to it, to which it is not uploading in return. If it finds one that is uploading at a rate higher than the peer it just choked, it will unchoke it. Otherwise, it performs an optimistic unchoke.

C. Variable Number of Outgoing Connections

This modification, denoted VOC, is a more significant departure from the BitTorrent protocol. Instead of all peers having a fixed number of outgoing connections, the number of connections a peer attempts to maintain depends on its upload capacity.

A simple approach is to set the upload rate for each connection to the same value for all peers, fixing it at some rate r_f. Therefore, if a peer has an upload capacity of r_c, it establishes $k = \left\lfloor \frac{r_c}{r_f} \right\rfloor$ connections. However, with this approach a peer wastes $r_c \mod r_f$ of its capacity. Thus, a better choice is to have any given peer upload at a rate of $r_f + \frac{r_c \mod r_f}{k}$. This means there will be some variability in the upload rates of different peers, but each rate is assured to be at least r_f.

The basic idea behind this approach is that any pair of peers can establish a connection between one another in which the individual upload rates are nearly identical irrespective of the discrepancy between peer upload capacities. For example, a high capacity peer might establish connections to twenty low capacity peers, and exchange data with each in a fair manner, whereas a low capacity peer might only maintain two connections. A pair of peers is allowed to have multiple connections between each other. This is particularly important for enabling pairs of high capacity peers to transmit data to one another at high rates.

We propose that each peer evaluate its set of outgoing and incoming connections every 10 seconds. At each iteration, it makes a list L_{nd} of peers to which it is currently uploading, but from which it is not receiving any data. It immediately chokes all of these peers. Next, it makes a list L_{nu} of peers from which it is downloading, but to which it is not uploading. If $|L_{nu}| \geq |L_{nd}|$, it begins uploading to a random set of $|L_{nd}|$ peers in L_{nu}. If $|L_{nu}| < |L_{nd}|$, it begins uploading to all peers in L_{nu}, and optimistically unchokes $|L_{nd}| - |L_{nu}|$ additional peers chosen at random.

V. Results

In this section we present the results generated via our simulator. In all cases, we consider a network with $N = 100$ peers over a 1-hour interval. For the Multiple Connection Chokes modification we set the Threshold Ratio to 1.1, and MC to 3. We determined experimentally that these values appear to provide the best performance (although the results for Threshold Ratios in the range 1.1–1.3 and MC from 2–3 are similar). We do not claim that these two values are always the optimal choice, which is probably dependent on the distribution of peers' upload capacities. For the VOC modification, we set r_f to a normalized upload rate of 0.025, as this ensures that, with the chosen upload capacity distribution, each peer will have at least 2 outgoing connections. Theoretically, using an extremely small value of r_f produces the best fairness because it results in negligible differences between different peers' upload rates. However, there is overhead associated with each connection and it is impractical to set r_f to an excessively small value.

We define the Time-Averaged Fairness Ratio (TAFR) for a particular peer as the ratio of data uploaded to data downloaded, averaged over the entire hour. We also introduce the *Average Ranking Difference* (ARD): Peers are ranked from lowest to highest upload capacity, and the *Ranking Difference* (RD) for any current connection is the absolute value of the difference between the rank of the uploading peer and that of the downloading peer. The ARD at any point in time is then defined as the the average RD of the 500 current upload sessions. It is easy to verify that the ARD of the equilibrium state described in Section III is $3\frac{8}{9}$ (if one ignores the peers in the lowest ranked group). Thus, we assert that the difference between the steady-state ARD of a scenario and the theoretical lower limit of approximately 4 gives a good indication as to how close to the "ideal" case the current set of peer connections is.

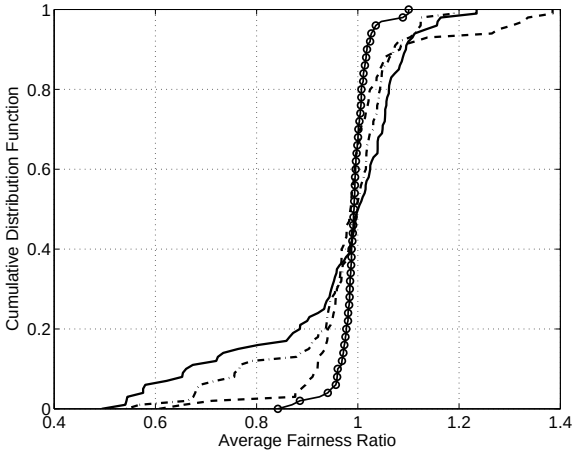
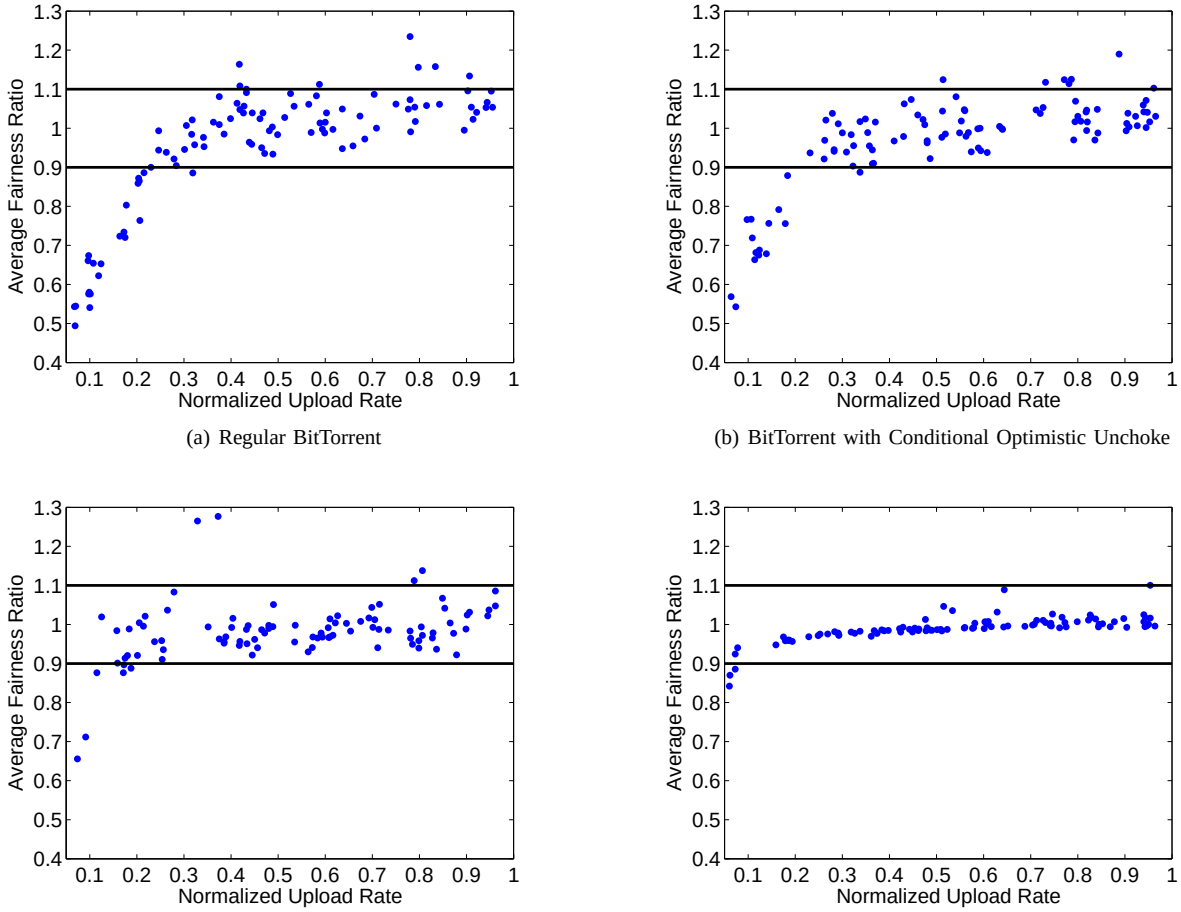


Fig. 1. Empirical Cumulative Distribution Function of Average Fairness Ratio. The solid curve is for regular BitTorrent, the alternating dashed and dotted curve represents BitTorrent with Conditional Optimistic Unchoke, the dashed line is for BitTorrent with Multiple Connection Chokes, and the line with circle-markers corresponds to BitTorrent with VOC.

Figure 1 presents the empirical Cumulative Distribution Function of the TAFR for regular BitTorrent and the three proposed modifications. Figure 2 includes scatterplots of the TAFR versus upload capacity for the 100 peers for the four cases. For regular BitTorrent, peers with low upload capacities tend to download disproportionately more data than they provide to other peers. This is attributable to the BitTorrent optimistic unchoke mechanism: probabilistically, most peers that randomly choose to upload to a low capacity peer will have a higher upload capacity. Although these peers will typically choke this new upload session quickly, after determining that the low capacity peer cannot offer a comparable upload rate in return, the BitTorrent protocol ensures that data is transferred for at least 10 seconds. Figure 2 illustrates that low bandwidth peers are randomly chosen by other peers at a high enough average rate to enable them to download more data than they upload. Conversely, high capacity peers tend to upload more than they download. Again, this can be attributed to the optimistic unchoke mechanism: when a high capacity peer chooses another peer at random, the majority of the time this will be a peer with significantly lower upload rate.

The Conditional Optimistic Unchoke modification introduces a marginal improvement in the TAFR distribution, as is best illustrated by Figure 1. The Multiple Connection Chokes modification significantly reduces the number of peers with a TAFR of less than 0.85, indicating that this modification



(a) Regular BitTorrent

(b) BitTorrent with Conditional Optimistic Unchoke

(c) BitTorrent with Multiple Connection Choke

(d) BitTorrent with Variable Number of Outgoing Connections.

Fig. 2. Scatterplots of Average Fairness Ratio versus Upload Capacity.

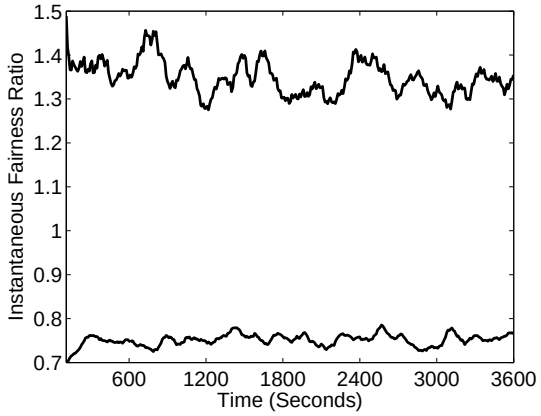
reduces the unfair advantage that peers with low upload capacities enjoy under regular BitTorrent. This is because it allows a high capacity peer to terminate a connection to a low capacity peer earlier. Finally, the VOC modification provides excellent fairness. Approximately 90% of peers have a TAFR between 0.95 and 1.05.

Figure 3 shows the average Instantaneous Fairness Ratio averaged over all the peers in the network. The two curves correspond to peers with an IFR of less than 1 and more than 1. For regular BitTorrent, there is a slight trend toward improvements for approximately the first 600 seconds, at which point the system appears to fluctuate about a steady-state. The Conditional Optimistic Unchoke modification displays improvement for approximately 1000 seconds, and at a higher rate. The Multiple Connection Choke modification continues to show an improvement for about about 1200 seconds, and achieves even better fairness. Finally, with VOC the system rapidly converges and shows the best steady-state fairness. We note that in steady-state, some of upper IFR curves take on larger values than the maximum TAFR any peer takes on in Figure 2. The reason for this is that the IFR of any given peer may vary a signficant amount over time: a peer included in the upper IFR curve at a certain point in time may quickly lower its IFR and be included in the lower IFR shortly thereafter.

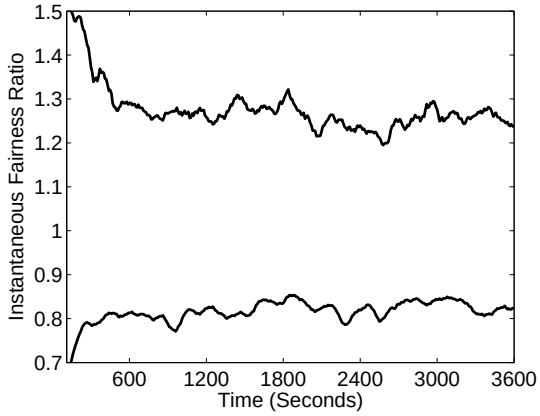
Figure 4 illustrates the Average Ranking Differences. We note that the relative steady-state ARD rankings mirror those of the three protocols' IFR and TAFR. Furthermore, the amount of time during which the ARD decreases for each case corresponds approximately to the duration during which the IFR improves. This provides evidence that ARD is indeed a relevant measure of performance.

VI. Conclusion

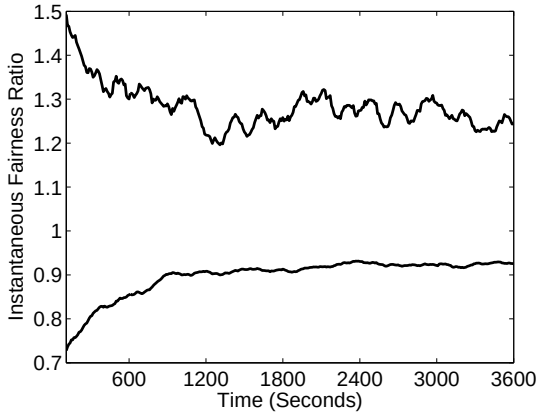
We have presented three modifications to the BitTorrent protocol intended to improve its fairness. According to our simplified model of BitTorrent, all three provide some level of improvement. The rankings, in order of increased improvement to fairness, are Conditional Optimistic Unchoke, Multiple Connection Choke, and Variable Number of Outgoing Connections. This order also corresponds to how radically each proposal modifies the BitTorrent protocol, and thus likely the degree of difficulty in practical implementation. In future work, we will assess these modifications using more accurate simulations of the BitTorrent protocol and network behaviour.



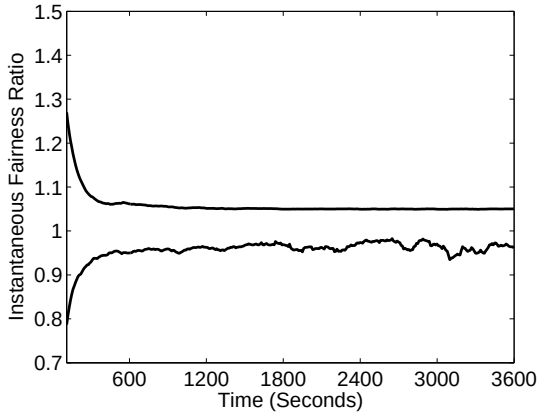
(a) Regular BitTorrent



(b) BitTorrent with Conditional Optimistic Unchoke.



(c) BitTorrent with Multiple Connection Choke



(d) BitTorrent with Variable Number of Outgoing Connections.

Fig. 3. Average Instantaneous Fairness Ratio versus time, over all peers. For each plot the upper curve is the average IFR over all peers with an IFR greater than 1, and the lower curve is the average IFR for peers with an IFR less than 1.

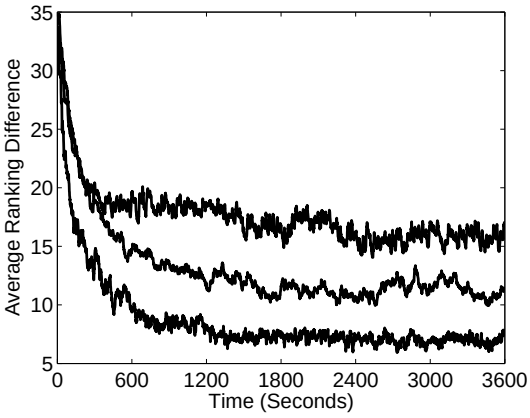


Fig. 4. Average Ranking Difference (ARD) vs. Time. From top to bottom, the three curves are for regular BitTorrent, BitTorrent with Conditional Optimistic Unchoke, and BitTorrent with Multiple Connection Chokes. The ARD for BitTorrent with VOC is not shown, as it is not a relevant measure of performance for this case.

REFERENCES

[1] M. Izal, G. Urvoy-Keller, E.W. Biersack, P. Felber, A. Al Hamra, and L. Garcés-Erice, "Dissecting BitTorrent: Five months in a torrent's lifetime," in *Proc. Passive Analysis and Measurement Workshop*, Antibes, Juan-les-Pins, France, April 2004.

[2] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips, "The BitTorrent P2P file-sharing system: Measurements and analysis," in *Proc. Int'l Workshop on Peer-to-Peer Systems*, Ithaca, NY, Feb. 2005.

[3] A. Bharambe, C. Herley, and V. Padmanabhan, "Understanding and deconstructing BitTorrent performance," Tech. Rep. MSR-TR-2005-03, Microsoft Research, 2005.

[4] "BitTorrent," `http://bittorrent.com`.

[5] B. Cohen, "Incentives build robustness in BitTorrent," `http://bittorrent.com/bittorrentecon.pdf`, 2003.

[6] "Edonkey2000," `www.edonkey2000.com`.

[7] "Gnutella protocol development," `http://rfc-gnutella.sourceforge.net/`.

[8] A. Bharambe, C. Herley, and V. Padmanabhan, "Understanding and deconstructing BitTorrent performance," in *Proc. ACM Sigmetrics*, Banff, Alberta, Canada, June 2005, short paper.

[9] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. ACM Sigcomm*, Portland, OR, Aug. 2004.