This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2011

Reinforcement Learning in BitTorrent Systems

Rafit Izhak-Ratzin
Palo Alto Networks
Sunnyvale, CA
rratzin@paloaltonetworks.com

Hyunggon Park†
Department of Electronics Engineering
Ewha Womans University, Seoul, Korea
hyunggon.park@ewha.ac.kr

Mihaela van der Schaar
Electrical Engineering Department
University of California Los Angeles, CA
mihaela@ee.ucla.edu

***Abstract*— In this paper, we propose a BitTorrent-like protocol that replaces the peer selection mechanisms in the regular BitTorrent protocol with a novel reinforcement learning based mechanism. The inherent operation of P2P systems, which involves repeated interactions among peers over a long time period, allows peers to efficiently identify free-riders as well as desirable collaborators by learning the behavior of their associated peers. Thus, it can help peers improve their download rates and discourage free-riding (FR), while improving fairness. We model the peers' interactions in the BitTorrent-like network as a repeated interaction game, where we explicitly consider the strategic behavior of the peers. A peer that applies the reinforcement learning based mechanism uses a partial history of the observations on associated peers' statistical reciprocal behaviors to determine its best responses and estimate the corresponding impact on its expected utility. The policy determines the peer's resource reciprocations with other peers, which would maximize the peer's long-term performance.**

***Index Terms*— P2P, BitTorrent, reinforcement learning**

I. Introduction

Peer-to-peer (P2P) content sharing protocols dominate the traffic on the Internet, and thus, have become an important part in building scalable Internet applications [1].

In P2P content distribution systems, fairness among peers is an important factor, because it encourages peers to actively collaborate in disseminating content, thereby improving system performance. However, even BitTorrent (BT) [2], one of the most popular P2P protocols, does not provide fair resource reciprocation, particularly for node populations having heterogeneous upload bandwidth [3]–[5]. This is because the tit-for-tat strategy implemented in BT only exploits a short-term history for making upload decisions. More specifically, upload decisions are made based on the most recent observations of the resource reciprocation. This also implies that the upload decisions are backward-looking and not forward-looking. Thus, a peer can keep following the tit-for-tat policy only if it continuously uploads pieces of a particular file to its associated peers and as long as it receives pieces of interest in return. However, this is not always feasible, since irrespective of peers' willingness to cooperate, they may not always have pieces that are of interest to the other peers [6]. Yet, such behavior is still perceived as a lack of cooperation for interacting peers. Moreover, it has been shown that BT systems do not effectively cope with peers' non-cooperative behaviors such as free-riding (FR) [7]–[9], because of their optimistic unchoke mechanism.

In this paper, we model the peers' interactions in the BT-like network as a repeated interaction game – repeated interactions (i.e., reciprocating resources) among several players (i.e., peers) in which a player takes actions (i.e. unchoke/choke peers) so as to maximize its long-term reward (i.e., cumulative download rates) [10]. The underlying state of the environment changes stochastically, and is contingent upon the decisions of the players. In our model, peers can adopt the reinforcement learning (RL) [11] algorithm to make their upload decisions. The peers in the network have partial information history about the reciprocation behaviors of their associated peers. Based on this information, the peers applying the RL-based strategy can estimate their future expected rewards, and then, can determine their best responses accordingly. The RL algorithm allows peers to improve their peer selection strategies based solely on the knowledge of their past interactions. It enables each peer to forecast the impact of the current peer selection on the future expected utility which it tries to maximize.

The proposed protocol can replace the optimistic unchoke, which is the most vulnerable process that allows free-riders [7], [9], with the RL-based unchokes. This improves the system performance. Moreover, the BT systems relying on short-term history may suffer from the lack of fairness, as pointed out in e.g., [3], [4], [12]. The proposed approach, however, can also improve the fairness by using a strategy based on a long-term history. As discussed in [13], there is a fundamental tradeoff between performance and fairness in BT-like protocols. We study the tradeoff in the proposed approach and show that the RL-based strategy improves the fairness of the system while penalizing the low-capacity leechers by reducing their download rates. By providing better incentives, the proposed approach can encourage peers to contribute more resources. The importance of contribution incentives in P2P systems is widely studied and different alternatives are proposed (see e.g., [3], [14]). To the best of our knowledge, we are the first to propose the RL-based strategy that can replace the existing mechanisms deployed in BT protocol, while maximizing long-term utility of participating leechers.

The proposed protocol is implemented on top of an actual BT client, and has been evaluated through extensive experiments in a controlled Planetlab testbed. The RL-based mechanism is executed simply through policy modifications to existing clients with no changes to the BT protocol. Our protocol does not demand full or sparse adoption of the RL-based peer selection mechanism (as in [3]) and can be run

†Corresponding author.

This work was supported by NSF CAREER Award CCF-0541867, Swiss National Science Foundation grants 200021-118230, and Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (2010-0009717).

by any number of peers. Based on the experimental results, several advantages of the proposed protocol against the regular BT protocol are summarized as: i) it discourages FR by limiting the upload to non-cooperative peers, ii) it promotes cooperative resource reciprocation among high-capacity peers, iii) it improves fairness by increasing (decreasing) download rates to the peers that contribute more (fewer) resources, iv) it improves the system robustness by minimizing the impact of FR on the contributing peers' performance, and v) it improves the stability of the peer selection mechanism, which affects directly the performance of the system.

II. Reinforcement Learning for Resource Reciprocation in P2P Networks

Peers in BT-like systems often make repeated decisions to select unchoked peers given their dynamically changing environment. The evolution of the peers' interactions across the various rechoke periods is modeled as a repeated interaction game [10].

In each rechoke period, every peer chooses its optimal action given the state it is in, which is represented by the set of resources received from its associated peers. The peers choose their own actions independently and simultaneously. After that, the peers are rewarded (i.e., total download rates) for taking their actions and transit into the next states. The reward and the state transition depend on the other peers' states and actions. During the repeated interactions of multiple peers, the peers can only observe a partial history of their associated peers' reciprocation behaviors. Based on these observations, the peers can estimate their future expected rewards and can identify their best responses (or best actions).

We adopt the RL to estimate the future peer's expected reward, as it leads to improving the peer selection strategy using only knowledge of the peer's own past reciprocation. Formally, an RL environment can be represented by a tuple, $\langle \mathbf{I}, \mathcal{S}, \mathcal{A}, P, R \rangle$. $\mathbf{I}$ is a set of peers in the game, $\mathbf{I} = \{1, \ldots, M\}$ for M peers in the game. $\mathcal{S}$ is the set of state profiles of all peers in the game, i.e., $\mathcal{S} = \mathbf{S}_1 \times \cdots \times \mathbf{S}_M$, where $\mathbf{S}_j$ is the state space of peer j. $\mathcal{A}$ is the joint action space $\mathcal{A} = \mathbf{A}_1 \times \cdots \times \mathbf{A}_M$, where $\mathbf{A}_j$ is the action (peer selection) space for peer j. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is a state transition probability function that maps from state profile $\mathcal{S}(t) \in \mathcal{S}$ at time t into the next state profile $\mathcal{S}(t+1) \in \mathcal{S}$ at time $t+1$ given corresponding joint actions $\mathcal{A}(t) \in \mathcal{A}$. Note that t here is discrete and measured in time slots. Finally, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+^M$ is a reward vector function defined as a mapping from the state profile $\mathcal{S}(t) \in \mathcal{S}$ at time t, and corresponding joint actions $\mathcal{A}(t) \in \mathcal{A}$, to a vector with each element being the reward to a particular peer.

To find the optimal policy in the game peers may require the entire history of the interactions among peers in the networks. However, this may be infeasible for real P2P networks. Unlike such games, finding an RL-based policy only requires the peers' own histories of observations through their experiences. The history of observations in the network up to time $(t-1)$ is defined as

$$\mathbf{H}(t) = \{\mathcal{S}(0), \mathcal{A}(0), \mathcal{R}(0), \ldots, \mathcal{S}(t-1), \mathcal{A}(t-1), \mathcal{R}(t-1)\}$$

which summarizes all previous states, actions and rewards of the peers in the network up to $(t-1)$. A peer j can only access a portion of the history of observations $\mathbf{H}(t)$, which is expressed as $\mathbf{O}_j(t) (\subseteq \mathbf{H}(t))$. The current state $\mathbf{S}_j(t)$ is always observable, i.e., $\mathbf{S}_j(t) \in \mathbf{O}_j(t)$. The state transition probability is calculated from $\mathbf{O}_j(t)$. This implies that the state transition probability considers long-term history of observations, which enables peers to capture their associated peers' behavior.

1) State Space of Peer j – $\mathbf{S}_j$: The state of peer j represents the set of resources received from the peers in C_j, where C_j denotes the set of peers associating with peer j. Thus, it may represent the uploading behavior of its associated peers, or equivalently, it can capture peer j's download rates from its associated peers. The upload rates from peer $i \in C_j$ to peer j at time t are denoted by $L_{ij}(t)$. In the proposed protocol, an uploading behavior of peer i observed by peer j is described by s_{ij}, and defined as $s_{ij} \in \{0, 1\}$, where $s_{ij} = 1$ if $L_{ij} > \theta_j$ and $s_{ij} = 0$ otherwise. θ_j is a pre-determined threshold of peer j. The state space of peer j can be expressed as

$$\mathbf{S}_j = \{ (s_{1j}, \ldots, s_{Nj}) | \, s_{ij} \in \{0, 1\} \text{ for all } i \in C_j \} \quad (1)$$

where N denotes the number of peer j's associated peers in C_j, i.e., $|C_j| = N$. Therefore, a state $\mathbf{S}_j(t) \in \mathbf{S}_j$ can capture the uploading behavior of the associated peers at time t.

2) Action Space of Peer j – $\mathbf{A}_j$: The action of peer j represents the set of its peer selection decisions. The action of peer j to peer i at time t is denoted by a_{ji}, and is defined as $a_{ji}(t) = \{0, 1\}$, where $a_{ji} = 0$ if peer j chokes peer i and $a_{ji} = 1$ otherwise. The action space of peer j can thus be expressed as

$$\mathbf{A}_j = \{ (a_{j1}, \ldots, a_{jN}) | \, a_{ji} \in \{0, 1\} \text{ for all } i \in C_j \}, \quad (2)$$

Hence, an action $\mathbf{A}_j(t) \in \mathbf{A}_j$ is of vector that consists of peer j's peer selection decisions to its associated peers at time t.

In the proposed protocol, we assume that peer j is able to unchoke $N_u (\leq N)$ peers, and the bandwidth allocated to an unchoked peer i by peer j at time t is determined by $L_{ji}(t) = B_j / N_u$, where B_j is peer j's maximum upload bandwidth[1].

3) State Transition Probability of Peer j: A state transition probability represents the probability that an action $\mathbf{A}_j(t) \in \mathbf{A}_j$ of peer j in state $\mathbf{S}_j(t) \in \mathbf{S}_j$ at time t will lead to another state $\mathbf{S}_j(t+1) \in \mathbf{S}_j$ at time $t+1$. This can be expressed as

$$P_{\mathbf{A}_j(t)}(\mathbf{S}_j(t), \mathbf{S}_j(t+1)) = \Pr(\mathbf{S}_j(t+1) | \mathbf{S}_j(t), \mathbf{A}_j(t)). \quad (3)$$

A peer j can estimate the state transition probability functions based on its history interactions of $\mathbf{S}_j(t')$, $\mathbf{A}_j(t')$ and $\mathbf{S}_j(t'+1)$ for $t' < t$, which can be stored in a transition table.

4) The Reward of Peer j – R_j: The peer's reward in a state is the sum of the estimated download rates from all of its associated peers. More specifically, a reward of peer j from state $\mathbf{S}_j(t) \in \mathbf{S}_j$ can be expressed as

$$R_j(\mathbf{S}_j(t)) = \left\langle \mathbf{S}_j(t), [L_{ij}]_{i \in C_j} \right\rangle = \sum\nolimits_{i \in C_j} L_{ij} \quad (4)$$

[1]While peer j allocates the same amount of upload bandwidths to all unchoked peers, the variable case can be future explored.

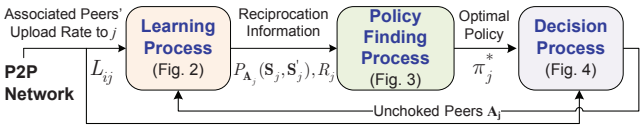

Fig. 1. Main processes in the proposed protocol design.

where $\langle \mathbf{X}, \mathbf{Y} \rangle$ denotes the inner-product between two vectors of $\mathbf{X}$ and $\mathbf{Y}$.

5) RL-based Policy π_j: The policy π_j obtained from the RL can provide a specific action $\mathbf{A}_j(t)$ for peer j in state $\mathbf{S}_j(t)$ at time t, i.e., $\pi_j : \mathcal{S}_j \rightarrow \mathcal{A}_j$. Thus, $\mathbf{A}_j(t) = \pi_j(\mathbf{S}_j(t))$. The actions are determined such that they maximize the cumulative discounted expected reward, defined for a peer j in state $\mathbf{S}_j(t)$ at time $t = t_c$ given a discount factor γ_j as

$$R_j^f(\mathbf{S}_j(t_c)) \triangleq \sum\nolimits_{t=t_c+1}^{\infty} \gamma_j^{(t-(t_c+1))} \cdot R_j(\mathbf{S}_j(t)). \quad (5)$$

Thus, the policy π_j maps each state $\mathbf{S}_j(t) \in \mathbf{S}_j$ into an action, i.e., $\mathbf{A}_j(t) = \pi_j(\mathbf{S}_j(t))$, such that each action maximizes $R_j^f(\mathbf{S}_j(t_c))$. The policy can be deployed as a peer selection algorithm, to enable peers to maximize their long-term utility.

III. The Protocol Design

In this section, we describe the proposed protocol design, which is summarized in Fig. 1. The protocol consists of three main processes running in parallel: *the learning process, the policy finding process, and the decision process*.

A. The Learning Process

The learning process provides updated information about statistical behaviors of the associated peers' resource reciprocation $\mathbf{O}_j(t)$. This process is necessary since the peers' reciprocation behaviors are not foretold. Therefore, peers are required to act in the environment in order to gather observation. In our proposed protocol, a peer adopting RL-based policy learns its environment statistically (i.e., the changes of states, rewards, etc., and the corresponding state transition probability) using the past observation it made about its associated peers. Thus, each peer needs to update the above information regularly through the learning process, while downloading content from its associated peers. The learning process consists of the following main methods (see Fig. 2.)

1) Computing Reward Method: The reward of peer j represents its total download rates from its associated peers estimated by peer j. In the reward calculation method, the associated peers are classified into two types - peers *with* and *without* reciprocation history based on the available information about their resource reciprocation history.

If peer j has a reciprocation history of peer i, it estimates the upload rates L_{ij} of peer $i \in C_j$ based on the weighted average of the recent upload rate samples L_{ij}^o, i.e.,

$$L_{ij}(t+1) \leftarrow \alpha_j \cdot L_{ij}^o(t+1) + (1-\alpha_j) L_{ij}(t) \quad (6)$$

where α_j is the weight for most recent resource reciprocation.

If peer j does not have a reciprocation history of peer i, it initializes the information about peer i by optimistically estimating that peer i reciprocates its resources with high probability and high upload rate. This enables peer j to efficiently discover additional peers and bootstrap newly joining peers, which is important for the system efficiency. Whenever peer j uploads to a peer without resource reciprocation history and the peer does not upload to j in return, peer j reduces the peer's presumed upload rate, as this provides j with more confidence that the particular peer may not actively reciprocate its data. This also prevents the associated peers from taking advantage of a peer through optimistic initialization and possible FR. Note that white-washing [15] is not possible in our design either, since peers are identified by their IP addresses.

2) Finding State Transition Probability Method : Each peer can capture the time-varying resource reciprocation behaviors of its associated peers by updating the state transition probabilities in every rechoke period. Every rechoke period at $(t+1)$, peer j stores 3-bit triplets for its associated peer i, $(s_{ij}(t), a_{ji}(t), s_{ij}(t+1))$. In our design, we assume that the state transition of each peer is independent. Thus, the state transition probability $P_{\mathbf{A}_j(t)}(\mathbf{S}_j(t), \mathbf{S}_j(t+1))$ from $\mathbf{S}_j(t) = (s_{1j}(t), \ldots, s_{Nj}(t))$ to $\mathbf{S}_j(t+1) = (s_{1j}(t+1), \ldots, s_{Nj}(t+1))$ given an action $\mathbf{A}_j(t) = (a_{j1}(t), \ldots, a_{jN}(t))$ can be expressed as

$$P_{\mathbf{A}_j(t)}(\mathbf{S}_j(t), \mathbf{S}_j(t+1)) = \prod\nolimits_{i=1}^{N} \Pr(s_{ij}(t+1) | s_{ij}(t), a_{ji}(t)).$$

B. The Policy Finding Process

The policy finding process computes the policy using RL. It runs in parallel with the learning process, while using the information obtained by the learning process. This process needs to be running during the entire downloading process as the changes of peers' reciprocation behaviors can result in the policies obtained in the previous time slots being outdated. This process is depicted in Fig. 3.

For practical implementation, it is critical to reduce the number of peers that a peer considers for reciprocation, as this process may require high computational complexity when the number of associated peers becomes large. Thus, this process begins with reducing the set of associated peers, and then, finds the policy π_j that maximizes the cumulative discounted expected reward in (5) in the reduced peer set. In order to reduce the peer set, peer j only selects the peers that can reciprocate their resources with higher probability and with higher upload rate. Hence, peer j computes the expected rewards $\hat{L}_{ij}$ from each peer $i \in C_j$, defined as

$$\hat{L}_{ij}(t+1) = L_{ij}(t) \times \Pr(i \rightsquigarrow j), \quad (7)$$

where $\Pr(i \rightsquigarrow j)$ denotes the probability of resource reciprocation with peer i. Then, peer j reduces its associated peer set by iteratively eliminating the peers with the smallest $\hat{L}_{ij}$ in its associated peer set. More details about the algorithm can be found in [16].

C. The Decision Process

The decision process determines the peer selection decisions based on the policy and the current state. It includes the initialization phase and the RL phase, shown in Fig. 4.

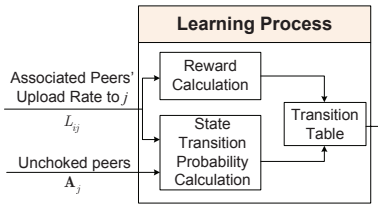

Fig. 2. The Learning Process

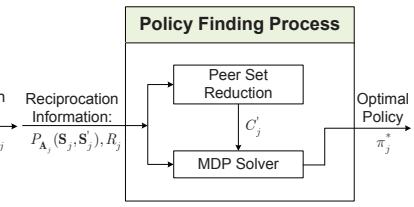

Fig. 3. The Policy Finding Process

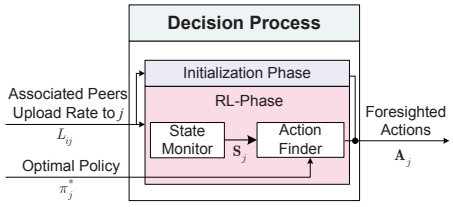

Fig. 4. The Decision Process

1) Initialization Phase: Since no information about associated peers is available for a newly joining peer j, peer j begins with adopting the regular BT mechanisms in the initialization phase. This enables peer j to collect reciprocation information with respect to its associated peers. During this phase, peer j discovers new peers, i.e., downloads from peers for the first time. Once j's peer discovery is slowed down (for more details, see [16]), it replaces the regular BT peer selection mechanisms, and operates in the RL phase.

2) RL Phase: In this phase, peer j determines the decisions on peer selection based on the policy obtained from the policy finding process in every rechoke period. Peer j determines its current state $\mathbf{S}_j$ and the corresponding action $\mathbf{A}_j$ (choke or unchoke), based on the policy π_j, i.e., $\mathbf{A}_j = \pi_j(\mathbf{S}_j)$.

IV. Experimental Evaluation

We perform extensive experiments on a controlled testbed, in order to evaluate the properties of the proposed protocol. We execute all the experiments consecutively in time on the same set of nodes. Unless otherwise specified, the default implementations of leecher and seed in regular BT systems are deployed. The upload capacities of the nodes are artificially set according to the bandwidth distribution of typical BT leechers [3]. All peers begin the download process simultaneously, which emulates a flash crowd scenario. The initial seeds have stayed connected through out the entire experiment. To provide synthetic churn with constant capacity, leechers disconnect immediately after completion of downloading the entire video file, and reconnect as new comers immediately while requesting the entire video file again. Unless otherwise specified, the experiments host 104 Planetlab nodes, 100 leechers and 4 seeds with a combined capacity of 128 KB/s, serving a 99 MB video file.

We implemented an RL-based client on top of the *Enhanced CTorrent* client, version 3.2 [17]. Our client can operate either in *RL-enhanced* mode using RL-based mechanisms, or in *regular mode* using the regular BT peer selection mechanism, for resource reciprocation. While only the main results of our experiments are highlighted in this paper due to the page limit, more details of the implementation such as the protocol prototype, design parameters, etc. and more extensive experiment results from various scenarios can be found in [16].

A. Single Leecher Adopting RL-enhanced Protocol

In this experiment, a single leecher adopts the *RL-enhanced* protocol, while the rest of the leechers run with the regular BT in a network having no free-rider (this is a common scenario that was tested by other proposed protocols such as [3], [14]).

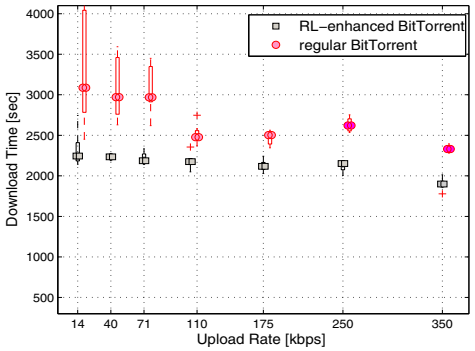

Fig. 5. Leecher' download time

Fig. 5 compares the download time of a single leecher, while adopting the *RL-enhanced* protocol and the regular BT protocol as a function of the leecher's upload capacity over seven trials. In this figure, separate boxplots are depicted for the different scenarios. The top and the bottom of the boxes represent the 75th and the 25th percentile sample of download time, respectively. The markers inside the boxes represent the median values, while the vertical lines represent the maximum and minimum of download time samples. Outliers are marked individually with "+" mark. We can observe that leechers having higher and lower capacities benefit from the *RL-enhanced* with 12%-27% improvement of their download time as indicated by the median value. This improvement provides leechers with an incentive to adopt the proposed protocol. By selecting to unchoke peers based on historical behavior information, our proposed protocol avoids the randomization that is presented in the regular BT tit-for-tat and optimistic unchoke implementations, which results in unstable peer selections and slow convergence.

B. RL-Enhanced Network without Free-Riders

We compare a system consisting of all leechers adopting the regular BT protocol with a system consisting of all leechers running in *RL-enhanced* mode. In this section, the network is without free-riders and 50 leechers are hosted. Fig. 6 shows the download completion time of leechers.

For each group of leechers that has the same upload capacity, separate boxplots are depicted for the different scenarios. The results clearly show the performance difference among high-capacity leechers, which are the fastest 20% leechers, and low-capacity leechers, which are the slowest 80% leechers. High-capacity leechers can significantly improve their download completion time – leechers having the upload capacity

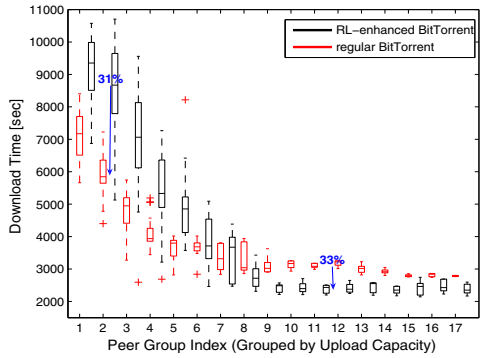

Fig. 6. Download completion time for leechers.

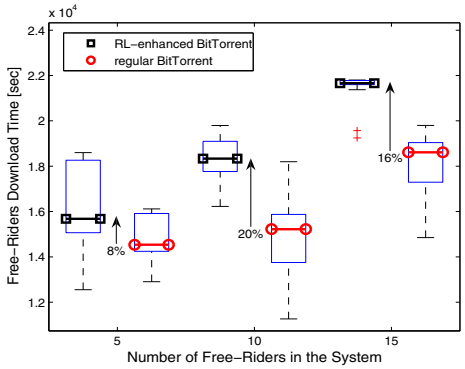

Fig. 7. Download completion time for free-riders.

of at least 18kB/sec improve their download completion time by up to 33% in median value. Unlike in the regular BT system, where leechers determine their peer selection decisions based on the tit-for-tat (myopic), the *RL-enhanced* leechers determine their peer selection decisions based on the long term history (foresighted). This enables the leechers to estimate the behaviors of their associated peers more accurately. Unlike regular BT, the random decisions of peer selection are also significantly reduced in the proposed approach, as the random decisions are taken only in the initialization phase or in order to collect the reciprocation history for newly joined peers. As a result, the high-capacity leechers can increase the probability of reciprocating resources with other high-capacity leechers.

This result also shows that fairness is improved in the *RL-enhanced* network, as high-capacity leechers can increase their download rates while the download rates of low-capacity leechers decrease compared to the regular BT system. Note that, however, all the peers that are slowed down by the RL-based strategy still download faster than their upload rates.

C. RL-Enhanced Network with Free-Riders

In this section, we investigate how effectively the proposed protocol can prevent selfish behaviors such as FRs. Fig. 7 shows the time that the free-riders complete downloading 99MB video file in a network consisting of 50 contributing leechers with several free-riders (i.e., 5, 10, and 15 free-riders). This result confirms that in the *RL-enhanced* network the leechers are able to effectively penalize the free-riders, as it takes longer time for the free-riders to complete their downloads e.g., 8%-20% more time as measured by the median value, in comparison to the regular BT protocol. This is because the *RL-enhanced* leechers can efficiently capture the selfish behaviors of the free-riders. Thus, they unchoke the free-riders with a significantly lower probability. Alternative set of simulation results in [16] shows that the leechers in the regular BT network upload approximately 2.8-3.7 times more data to the free-riders compared to the *RL-enhanced* network. Therefore, we can conclude that the *RL-enhanced* networks are more robust to the selfish behaviors of peers than the networks operating with the regular BT protocol.

V. Conclusion

We propose a BT-like protocol that can replace the peer selection mechanisms of the regular BT protocol with a novel RL-based mechanism. The resource reciprocations among peers are modeled as repeated interactions in a game. Using RL, peers can estimate the impact of associated peers' reciprocation behaviors on the expected future reward and can improve their reciprocation strategies. Experiment results based on our actual implementation show that the proposed protocol improves the stability of the peer selection mechanism, improves collaboration among high capacity peers, improves fairness, enhances the robustness of the network against non-cooperative behaviors such as FR, and improves the download rates of the peers deploying the protocol.

References

[1] IPOQUE Internet measurements 2008-2009. [Online]. Available: http://www.ipoque.com/
[2] B. Cohen, "Incentives build robustness in BitTorrent," in *P2PEcon*, 2003.
[3] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in BitTorrent?" in *NSDI*, 2007.
[4] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis, and modeling of BitTorrent-like systems," in *IMC*, 2005.
[5] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and sharing incentives in BitTorrent systems," in *SIGMETRICS*, 2007.
[6] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," in *NSDI*, 2008.
[7] N. Liogkas, R. Nelson, E. Kohler, and L. Zhang, "Exploiting BitTorrent for fun (but not profit)," in *IPTPS*, 2006.
[8] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *SIGCOMM*, 2004.
[9] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free riding in BitTorrent is cheap," in *HotNets-V*, 2006.
[10] H. Park and M. van der Schaar, "A framework for foresighted resource reciprocation in P2P networks," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 101–116, Jan. 2009.
[11] J. Hu and P. Wellman, "Multiagent reinforcement learning: Theoretical framework and an algorithm," in *ICML*, 1998.
[12] R. Izhak-Razin, "Collaboration in BitTorrent systems," in *Networking*, 2009.
[13] B. Fan, D.-M. Chiu, and J. C. S. Lui, "The delicate tradeoffs in BitTorrent-like file sharing protocol design," in *ICNP*, 2006.
[14] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, "BitTorrent is an auction: analyzing and improving BitTorrent's incentives," *SIGCOMM*, 2008.
[15] M. Feldman, C. Papadimitriou, and J. Chuang, "Free-riding and whitewashing in peer-to-peer systems," in *PINS*, 2004.
[16] R. Izhak-Ratzin, H. Park, and M. van der Schaar, "Foresighted resource reciprocation strategy in BitTorrent systems," UCLA - Computer Science Department, Tech. Rep. UCLA/CSD TR-10-0014, 2010.
[17] Enhanced CTorrent. [Online]. Available: http://www.rahul.net/dholmes/ctorrent