Signal Processing: *Image Communication* 27 (2012) 545–554

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

# A game theoretic approach to video streaming over peer-to-peer networks ☆

Ehsan Maani [a], Zhaofu Chen [b], Aggelos K. Katsaggelos [c,*]

[a] *Multimedia Research Lab, Sony Electronics Inc., San Jose, CA, USA*
[b] *Electrical Engineering and Computer Science Department, Northwestern University, M478, 2145 Sheridan Road, Evanston, IL 60208, USA*
[c] *Electrical Engineering and Computer Science Department, Northwestern University, M470, 2145 Sheridan Road, Evanston, IL 60208, USA*

ARTICLE INFO

Available online 16 February 2012

*Keywords:*
P2P multimedia sharing
Game theory

ABSTRACT

We consider the problem of foresighted multimedia resource reciprocation in peer-to-peer (P2P) networks, which consist of rational peers aiming at maximizing their individual utilities. We introduce an artificial currency (credit) to take into account the characteristics of different parts of the video signal. The resource reciprocation with the proposed credit metric can be formulated as a stochastic game, in which the peers determine their optimal strategies using Markov Decision Process (MDP) framework. The introduced framework can be applied to the general video coding, and in particular, is suitable for the scalable video where various parts of the encoded bit stream have significantly different importance for the video quality.



## 1. Introduction

With the rapid advancement in network technologies, multimedia services, and especially video-related services, including IPTV, broadcast of real-time sports, video-on-demand (VoD), etc., have gained great popularity and hence have become a significant contributor to today's Internet traffic.

Multimedia services, such as IPTV, broadcast of sports and events, video-on-demand (VoD), and multimedia sharing have become increasingly popular and contribute significantly to today's Internet traffic. Many of these services require media streaming to a large number of subscribers. Being specifically designed for such applications, IP Multicast is the most efficient vehicle for these services. However, lack of incentives to install multicast-capable routers to handle multicast traffic, due to many political and economical issues, has prevented IP Multicast from being widely deployed. Traditional server-based solutions are often used for multimedia content delivery in such applications. However, server-based solutions are very cost inefficient and have serious scaling issues. Therefore, especially in recent years, there has been significant interest in the use of peer-to-peer (P2P) technologies for Internet multimedia services. P2P solutions are highly cost efficient since they do not require special network infrastructure support and thus are easy to deploy. Furthermore, P2P approaches scale very well with large number of subscribers since greater demand usually brings greater resources along.

These characteristics also make P2P solutions extremely attractive for file sharing and on-demand media streaming. P2P file sharing solutions such as BitTorrent [2] are extremely popular and are responsible for a large percentage of today's Internet traffic. Moreover, several multimedia streaming systems have been successfully developed for P2P networks. These systems, based on the technologies they use, can be categorized into tree-based or data-driven approaches [3,4]. The vast majority

☆ An earlier version of this paper was presented at the 2010 *IEEE* International Conference on Image Processing (ICIP), Hong Kong [1].

* Corresponding author. Tel.: +1 847 491 7164; fax: +1 847 491 4455.
*E-mail addresses:* emaani82@gmail.com (E. Maani), zhaofuchen2014@u.northwestern.edu (Z. Chen), aggk@eecs.northwestern.edu (A.K. Katsaggelos).

0923-5965/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2012.02.015

of the existing research on P2P focuses on tree-based approaches, where peers are organized into fixed structures (i.e., trees) for delivering data. Nodes in the structure typically have well-defined relationships, e.g., "parent-child" relationships in trees. Parents are responsible for forwarding data packets they receive to their children. Consequently, the structure plays an essential role in the average user experience and thus has to be fully optimized. In addition, maintaining the tree structures has been shown to be very challenging since nodes can join and leave the group at any time. This maintenance is especially crucial for time-constrained content. In contrast to tree-based approaches, data-driven approaches do not require constructing and maintaining an explicit structure; instead, data availability is used to guide the data flow. In this work, we focus on data-driven P2P systems such as CoolStreaming [5], GnuStream [6], PeerStreaming [7] and Chainsaw [8] for multimedia streaming, or BitTorrent systems [2,9], for general file sharing.

In these systems, data (i.e., multimedia/graphics streams or general files) are divided into smaller pieces which are distributed over the network, thus, allowing multiple people to also upload data to others while they are streaming/downloading. Nodes periodically exchange data availability information with a set of partners to notify them of the pieces they have available. While this approach has been already shown to successfully work for P2P multimedia streaming, two important aspects of this have been rarely addressed in the existing literature: (1) the optimal resource reciprocation policy for self-interested peers to maximize their individual utility and perhaps build incentives in the network for sharing more resources; (2) the role of the multimedia signal characteristics in deriving the optimal policy to enhance quality-of-service (QoS).

One of the most popular resource reciprocation strategies, the tit-for-tat (TFT) policy is deployed in the BitTorrent protocol. A peer in BitTorrent systems equally divides its available upload bandwidth among multiple leechers who contribute the most to its download [2]. However, this policy is based on the assumption of equal upload bandwidth distribution, and thus, is not optimal for heterogeneous content and diverse peers (with different upload to download ratios). An alternative resource reciprocation policy is introduced in [5] which is based on a heuristic scheduling algorithm and allows the peers to determine the suppliers of required pieces and select the peer with the highest bandwidth. A reciprocation algorithm is also proposed in [8] based on random piece selection.

The traditional assumption of the nature of the peers in a P2P network has been that the peers are altruistic and will follow the prescribed protocols without deviation (see for example [8]). However, this view has changed in many recent research works. Peers are assumed to be autonomous and rational [10] and are, thus, incentivized using economic principles [10,11]. Consequently, in order to take into account the complex interactions of the self-interested and rational peers, game theoretic approaches have been proposed [10–13]. For example, in [11] an incentive-based scheme is designed such that the benefits that a peer can draw from the system are linked to its contribution. Therefore, rational and strategic players are motivated to contribute more to the system in order to maximize their utilities. Peers' decisions in these P2P systems are considered to be myopic. A foresighted decision making solution is proposed in [14], in which, peers determine their resource distributions by explicitly considering the probabilistic behaviors of their associated peers. In [14], the resource reciprocation game is formalized as a Markov Decision Process (MDP) which enables foresighted decision making. This MDP framework is modified in [15] to take into account the limited ability of peers to characterize their resource reciprocation status. The relationship between the reciprocation complexity and the resulting utility is studied and an algorithm is developed to determine the tradeoff. In [16], a pricing mechanism is introduced to bring in incentives for sharing and hence to improve efficiency. These studies, however, do not take the multimedia characteristics, including the stringent delay constraints, into account and assumes that video quality is simply a function of the downloading rate. In addition, another drawback of such a system is its exponentially growing computational complexity that becomes unmanageable when the number of peers in the group becomes large.

In this work, we envisage a practical peer-to-peer system specifically designed for multimedia sharing. Similar to [14], peers are assumed to be self-interested, strategic players which make foresighted decisions in order to maximize their own video quality. An MDP is employed to obtain the optimal resource allocation policy based on the other peers' reciprocation history. However, an approximation method is introduced in order to reduce the computational burden of the optimization. Since different pieces of the bit stream may have significantly different impact on the video quality, a fictitious currency is introduced to allow peers to offer different prices for various parts of their multimedia content [1]. Furthermore, at the packet level, the problem of optimal packet and packet-request scheduling is addressed.

This paper is structured as follows. In Section 2 we provide a brief introduction to multimedia distribution systems in P2P networks. Section 3 proposes the content-aware resource measurement and discusses the fictitious currency used to quantify the importance of multimedia content. In Section 4 we elaborate on the resource reciprocation framework in detail and introduce the solution algorithm. Experimental results are presented in Section 5, and conclusions are drawn in Section 6.

## 2. Multimedia distribution in P2P networks

In order to efficiently address the high variability in demand for video quality and resources each peer contributes to the network, scalable video coding (SVC) has been used in this work. However, our formulation is general and any video coding scheme may be used. A video bit stream is called scalable when parts of it can be removed in a way that the resulting substream forms a valid bit stream representing the content of the original

with lower resolution and/or quality. SVC is a video transmission and storage system highly suitable for the modern heterogeneous networks. SVC is particularly an attractive solution in scenarios, such as P2P networks, where the bandwidth capabilities, system resources and network conditions are not known in advance. Unlike the scalable profile of most prior international coding standards, the newly developed Scalable Extension of the H.264/AVC [17] provides a superb coding efficiency, high bit rate adaptability, and low decoder complexity, and thus, is considered an attractive video coding solution for environments of interest here.

The core operations in a data-driven overlay network are very simple: every node periodically exchanges data availability information with a set of partners, and retrieves unavailable data from a partner or more, or supplies available data to partners. Partners are interested in each others' multimedia content. Multiple approaches exist on how these partnerships (or groups) are formed and maintained [5,8,18]. An example of this partnership is illustrated in Fig. 1.

In this approach neither the partnership, such as the one shown in Fig. 1, nor the data transmission directions are fixed. Instead, they are rather dynamically determined by data availability. More specifically, video bit streams are divided into multiple segments (each consisting of a set of network abstraction layer (NAL) units) and the availability of each segment in the buffer of a node is represented by a buffer map (BM). Each node continuously exchanges its BM with partners, and then schedules which segments are to be fetched from which partners accordingly.

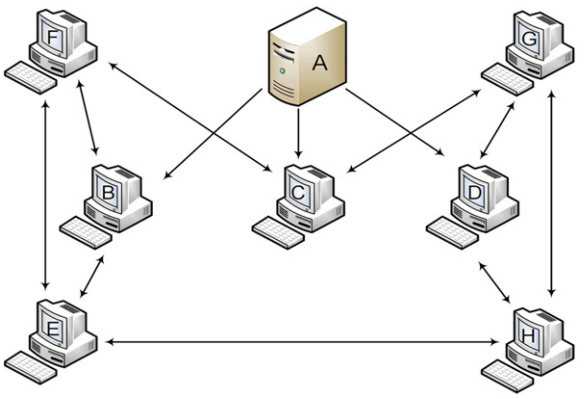


**Fig. 1.** An illustration of a group with A being the source node.

In some P2P file sharing solutions such as BitTorrent [2], files are divided into many equal-sized "pieces" shared by the interested peers. This enables the tracker to easily keep tabs on who has which piece of data and also facilitates the use of hash codes in .torrent files. The order in which these file pieces are received does not matter as long as all pieces of the files are received at the end of the session. However, unlike files in a file sharing application, various segments of multimedia data have significantly different importance and playback deadlines making the segmentation of the bit stream very difficult. Segmenting a video bit stream into equally sized pieces drastically reduces the efficiency of the system. As a result, a different approach is often proposed for multimedia sharing applications. A sliding window indicating the active buffer portion [3] is widely used in P2P multimedia streaming solutions including this work. For example, a sliding window of 120 segments, each consisting of 1-s video, is adopted in [5]. A BM thus consists of a bit-string of 120 bits, each indicating the availability of the corresponding segment. The sequence number of the first segment in the sliding window is also recorded by another 2 bytes, which can be rolled back for extra long video programs. The concept of the sliding window and comparison with BitTorrent is illustrated in Fig. 2.

While the approach above solves the problem of unequal playout deadlines of the different segments, it still assumes all segments have equal impact on the video quality. This assumption does not hold in general and even more so for scalable video. Therefore, each peer needs a method for prioritizing the NAL units based on their impact on the overall quality of the video sequence. Using such prioritization, each peer can assign a priority identifier (PID) to each NAL unit before transmission starts [19]. Furthermore, we assume that a "piece" covered by the sliding window consists of $m$ group of pictures (GOPs). The value of $m$ can be determined by jointly considering the playback speed and computational time cost. In our video streaming experiment, the sliding window covering one piece of data consists of $m=4$ GOPs. Then all NAL units within a piece are classified into $K$

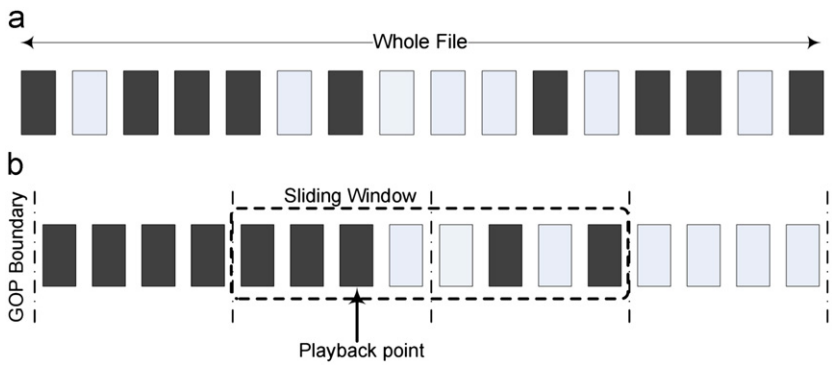


**Fig. 2.** (a) Buffer snapshots of BitTorrent and (b) the sliding window concept, where shaded segments are available in the buffer.

different classes according to their PIDs. Note that the computation of the PIDs can be done offline and PIDs can be recorded in the NAL unit headers [2]. Using the BitTorrent terminology, we refer to these sub-pieces (NALs that belong to various classes) as "blocks". Each block has a single PID which is the average of the PIDs of all the NAL units in the block (if they are different).

While streaming, peers can request a BM message from other peers in the group by specifying the region of the bit stream they are interested in. Then, the peers who receive a BM request message generate a BM message notifying the requesting peer of the pieces and blocks within the requesting window it has to offer. The BM message consists of 1 bit per piece to signal the piece availability, and if the piece is available, $\lceil \log_2 K \rceil$ bits for signaling the number of available blocks for that piece. Given the BMs of a node and its partners, a schedule is then generated for fetching the expected pieces and blocks from the partners. For each required block, a block request message is then sent to the appropriate partner (that has the block available). For a dynamic and heterogeneous network, the maximum streaming bandwidth of the partners has to be taken into account in the scheduling algorithm. Other peers in the group receiving requests for blocks decide to reply to the requests based on the playback deadlines of the blocks, their PIDs, and the amount of resources they want to provide to the requesting peer.

## 3. Content-aware resource measurement

Since various blocks have different impact on the quality of the reconstructed video, an incentive must be implemented within the system to encourage peers to prioritize and allocate more resources to the more important blocks. In order to provide that incentive, we introduce a fictitious currency called "credit". Depending on the requesting block's PID, a peer $i$ offers $\rho \in [1, +\infty)$ credits per bit to peer $j$ who replies to its request for the missing block. This agreed upon price is included in the block (packet) request. Therefore, peer $i$ will have to pay at some time in the near future after the acceptance of its request by peer $j$ for a missing piece. The exchange rate from bit to credit for each block $b_k$, i.e., $\rho(b_k)$, is a function of the block's PID. Each peer can arbitrarily choose the price for each of its block classes. However, a low price will result in its request being denied and a high price may cause failure to pay (based on peer's resources) which will be negatively reflected in its reciprocation history kept by peer $j$.

With the definition of the cost of a block, we can formulate the criteria by which peers reply to the requests made by other peers. Let $\mathcal{B}_j = \{b_{j:1}, b_{j:2}, \ldots, b_{j:n}\}$ be the set of all non-expired block requests made by peer $j$ in the buffer of peer $i$ (expired block requests are discarded). At any time $t$ peer $i$ decides on the packets to be sent to peer $j$ during the next time duration $\delta t$. This is done by sorting the requested packet set $\mathcal{B}_j$ based on the offered per bit prices, $\rho(b_{j:k})$ (signaled by peer $j$). Then, peer $i$ starts sending packets to peer $j$ until it reaches its pre-decided budget of $a_{ij}\delta t$ credits or the maximum number of bits it can upload. Let $u_{ij}$ and $a_{ij}$ denote the bit and credit rate peer $i$ is uploading to peer $j$ and $L(\cdot)$ the length of the packet in bits. Then the function $C_{ij}(u_{ij})$ mapping bit rate to credit rate is given by

$$a_{ij} = C_{ij}(u_{ij}) = \frac{1}{\delta t} \sum_{k=1}^{n^*} \rho(b_{j:k}) L(b_{j:k}), \tag{1}$$

where $n^*$ is the largest integer such that $\sum_{k=1}^{n^*} L(b_{j:k}) \le u_{ij}\delta t$ and the packets $b_{j:k}$ are ordered such that $\rho(b_{j:k}) > \rho(b_{j:k'})$ if $k < k'$. Eq. (1) shows how to compute the upload credit rate based on the upload bit rate $u_{ij}$. However, since resource reciprocation is measured in credits, peers first decide on the maximum number of credits they are willing to upload to others in the group and the actual bit rate is computed later based on that. Therefore, we also define the inverse of the function in (1) as

$$u_{ij} = C_{ij}^{-1}(a_{ij}) = \frac{1}{\delta t} \sum_{k=1}^{n^\dagger} L(b_{j:k}), \tag{2}$$

where $n^\dagger$ is the largest integer such that $\sum_{k=1}^{n^\dagger} \rho(b_{j:k}) L(b_{i:k}) \le a_{ij}\delta t$. Note that since the set of the requested packets $\mathcal{B}_j$ is known, peer $i$ can compute $C_{ij}^{-1}(a_{ij})$ for any value of $a_{ij}$.

## 4. Resource reciprocation framework

### 4.1. Resource reciprocation game

Resource reciprocation process in P2P networks can be modeled as a game environment in which the peers (players) are interested in obtaining multimedia contents from each other [14]. From the perspective of a single peer, the resource reciprocation game involves itself and a group of other peers associated with it. Examples of such a group include swarms [9,18], partnerships [5] and neighbors [8]. The group of peers associated with peer $i$ is denoted by $G_i$, which does not include peer $i$ itself. Any peer $k$ that belongs to group $G_i$, also has its own group $G_k$ which includes peer $i$. We denote the cardinality of the set $G_i$ by $n_i$, i.e., $n_i = |G_i|$. For a peer $i$, $\mathbf{A}_i$ is the action space, $P_i : \mathbf{S}_i \times \mathbf{A}_i \times \mathbf{S}_i \to [0, 1]$ is a state transition probability function that maps the state $\mathbf{s}_i \in \mathbf{S}_i$ at time $t$, the corresponding action $\mathbf{a}_i \in \mathbf{A}_i$ and the next state $\mathbf{s}_i' \in \mathbf{S}_i$ at time $t+1$ to a real number between 0 and 1, and $R_i : \mathbf{S}_i \to \mathbb{R}$ is a reward function, where $R_i(\mathbf{s}_i)$ is a reward derived in state $\mathbf{s}_i$. The details are explained as follows.

*State space* $\mathbf{S}_i$: A state of peer $i$ represents the set of discretized received resources (in credits per s) from the peers in $G_i$, i.e., $\mathbf{S}_i = \{\mathbf{s}_i = (s_{i1}, s_{i2}, \ldots, s_{in_i}), s_{ik} = Q(x_{ki}), \forall k \in G_i\}$, where $x_{ki}$ is the rate received by peer $i$ from peer $k$ and $Q(\cdot)$ is a quantization function.

*Action space* $\mathbf{A}_i$: An action of peer $i$ is its resource allocation (in credits per s) to the peers in $G_i$, i.e., $\mathbf{A}_i = \{\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in_i})\}$, where $a_{ik}$ denotes the allocated resources (also discretized) to peer $k$ by peer $i$. Note that peer $i$'s action $a_{ik}$ on peer $k$ becomes peer $k$'s received resources from peer $i$, i.e., $a_{ik} = s_{ki}$. An illustrative resource reciprocation is shown in Fig. 3.

*State transition probability* $P_{\mathbf{a}_i}(\mathbf{s}_i, \mathbf{s}_i')$: The transition between the current state $\mathbf{s}_i$ to a future state $\mathbf{s}_i'$ via an action $\mathbf{a}_i$ is governed by the state transition probability.

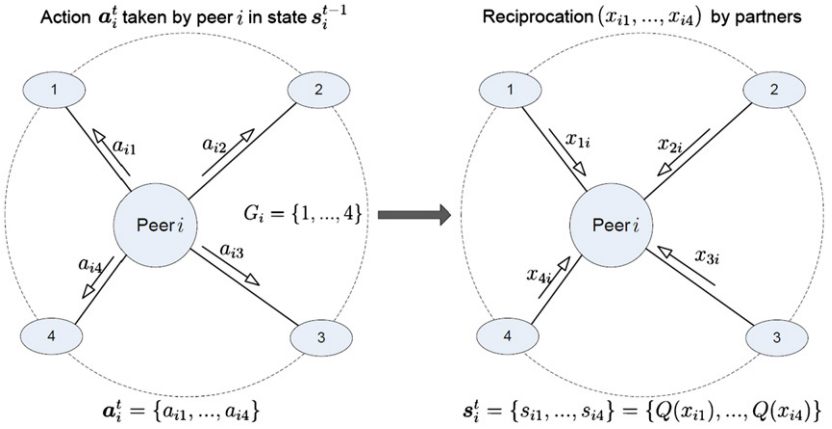


**Fig. 3.** An illustration of resource reciprocation in a group of 4 + 1 peers at time $t$.

Specifically, $P_{\mathbf{a}_i}(\mathbf{s}_i,\mathbf{s}_i')$ denotes the probability of state transfer from $\mathbf{s}_i$ to $\mathbf{s}_i'$ by taking action $\mathbf{a}_i$. Hence, given a state $\mathbf{s}_i \in \mathbf{S}_i$ at time $t$, an action $\mathbf{a}_i$ of peer $i$ can lead to another state $\mathbf{s}_i' \in \mathbf{S}_i$ at time $t' > t$ with probability $P_{\mathbf{a}_i}(\mathbf{s}_i,\mathbf{s}_i')$. These probabilities are learned online from the behaviors of the peers.

*Reward* $R_i$: The utility of peer $i$ downloading its desired multimedia content from other peers in $G_i$ is defined as the total credit downloading rate. If the credit rate received by peer $i$ from peer $j$ is denoted by $R_{ij}$ we have $R_i = \sum_j R_{ij}$. Since the distortion impact of the packets is reflected in their prices (i.e., denoted by $\rho_k$), the total number of received credits is a good measure of video distortion. As the packets are ordered according to their contribution to the video quality, the actual sequence distortion is a monotonically decreasing and convex function of the total credit rate.

*Reciprocation policy*: A policy $\pi : \mathcal{S} \to \mathcal{A}$ is a mapping from the states to the actions; thus, by executing a policy $\pi$ in state $\mathbf{s}$, the peer will take an action $\mathbf{a} = \pi(\mathbf{s})$. Each individual peer's ultimate goal is to maximize the sum of its discounted expected rewards to determine the best strategy, i.e.,

$$\max_{\mathbf{a}_i^{(0)},\mathbf{a}_i^{(1)},\ldots} E\left[\sum_{t=0}^{\infty} \gamma^t R_i^{(t)}(\mathbf{s}_i^{(t)},\mathbf{a}_i^{(t)})\right], \tag{3}$$

where $\gamma < 1$ and $R_i^{(t)}$, $\mathbf{s}_i^{(t)}$ and $\mathbf{a}_i^{(t)}$ indicate the reward, state and action of peer $i$ at a future time $t$. For each policy $\pi$, we define a *value function* as

$$V^{\pi} = E\left[\sum_{t=0}^{\infty} \gamma^t R_i^{(t)}(\mathbf{s}_i^{(t)},\pi(\mathbf{s}_i^{(t)}))|\mathbf{s}_i^{(-1)}\right], \tag{4}$$

where $\mathbf{s}_i^{(-1)}$ denotes the initial state. The value function $V^{\pi}$ satisfies the following Bellman equation

$$V^{\pi}(\mathbf{s}) = R_i(\mathbf{s}_i,\pi(\mathbf{s}_i)) + \gamma \sum_{\mathbf{s}_i' \in \mathbf{S}_i} P_{\mathbf{s}_i',\mathbf{s}_i}(\pi(\mathbf{s}_i)) V^{\pi}(\mathbf{s}_i'). \tag{5}$$

The optimal value function $V^*$ which corresponds to the optimal policy $\pi^*$ maximizing (3) can be computed using a well-known technique called value iteration [20]:

$$V^k(\mathbf{s}_i) = \max_{\mathbf{a}_i \in \mathbf{A}_i} R_i(\mathbf{s}_i,\mathbf{a}_i) + \gamma \sum_{\mathbf{s}_i' \in \mathbf{S}_i} P_{\mathbf{s}_i',\mathbf{s}_i}(\mathbf{a}_i) V^{k-1}(\mathbf{s}_i'), \tag{6}$$

where $k$ is the iteration index and $V^0$ is an arbitrary initial value function. $V^*$ is obtained as $k \to \infty$, and therefore, given this optimal value function, $\pi^*$ can be computed as

$$\pi^*(\mathbf{s}_i) = \arg\max_{\mathbf{a}_i \in \mathbf{A}_i} R_i(\mathbf{s}_i,\mathbf{a}_i) + \gamma \sum_{\mathbf{s}_i' \in \mathbf{S}_i} P_{\mathbf{s}_i',\mathbf{s}_i}(\mathbf{a}_i) V^*(\mathbf{s}_i'). \tag{7}$$

For each peer $i$ with an action space $\mathbf{A}_i$, the optimizations in (6) and (7) are subject to the user's upload speed $U_i$ at all times, i.e.,

$$\sum_{j \in G_i} u_{ij}^{(t)} = \sum_{j \in G_i} C_{ij}^{-1}(a_{ij}^{(t)}) \le U_i, \tag{8}$$

where $u_{ij}$ and $C_{ij}^{-1}$ are defined in the previous section. At each iteration, all possible combination of actions and future states for all peers of the group have to be considered together. More precisely, each iteration will have a computational complexity of $\mathcal{O}(|\mathbf{s}_i|^{3n_i})$ where $|\mathbf{s}_i|$ denotes the number of states per peer. Therefore, the problem becomes unmanageable quickly as the number of peers in the group grows.

To get over the curse of complexity, we consider each of the peer $j \in G_i$ separately while satisfying the overall constraint of (8):

$$\forall j \in G_i, \quad \max_{a_{ij}^{(0)},a_{ij}^{(1)},\ldots} E\left[\sum_{t=0}^{\infty} \gamma^t R_{ij}^{(t)}(\mathbf{s}_{ij}^{(t)},\mathbf{a}_{ij}^{(t)})\right]. \tag{9}$$

The constraint of (8) can be efficiently imposed utilizing the Quadratic Penalty method. Therefore, the value function for the $k$th iteration of the value iteration method can be expressed as

$$V^k(\mathbf{s}_i) = \max_{\mathbf{a}_i \in \mathbf{A}_i} \sum_{j \in G_i} V_j^k(s_{ij}) - \frac{\mu}{2}\left[\sum_{j \in G_i} C_{ij}^{-1}(a_{ij}) - U_i\right]^2, \tag{10}$$

where

$$V_j^k(s_{ij}) = R_{ij}(s_{ij},a_{ij}) + \gamma \sum_{s_{ij}' \in S_{ij}} P_{s_{ij}',s_{ij}}(a_{ij}) V_j^{k-1}(s_{ij}'). \tag{11}$$

The parameter $\mu$ controls the strength of the penalty due to deviation from the constraint. We observed that since the unconstrained optimization of (10) is not convex, steepest descent is a suitable and powerful technique to find its solution. The quadratic penalty strength $\mu$ is

initially small, at each iteration its value is increased, according to [21], to satisfy the constraint with a sufficient accuracy. Finally, the optimal policy is determined according to

$$\pi^*(\mathbf{s}_i) = \arg\max_{\mathbf{a}_i \in \mathbf{A}_i} \sum_{j \in G_i} V_j^*(s_{ij}) - \frac{\mu}{2}\left[\sum_{j \in G_i} C_{ij}^{-1}(a_{ij}) - U_i\right]^2, \quad (12)$$

where $V^*$ is obtained as $k \to \infty$. However, it should be noted that in practice the value function will not change significantly after a few iterations and therefore a number less than 10 is considered to be reasonable.

### 4.2. Packet request management

During transmission, when moving the sliding window forward, a peer needs to send out new packet requests for the missing packets within the sliding window. Thus, the newly emerged problem is how to efficiently allocate these new packet requests to various partners, such that video quality is maximized while all partners have the opportunity to increase their reciprocation level if they are willing to. Various uncertainty factors such as the many-body complex interactions of the peers in the group (e.g., the requested data rates from the peer's partners by other peers), and the dependencies within the video bit stream, make this problem quite challenging. The number of credits requested from a peer should be accurately determined by the peer's recent history. Excessive demand of a peer $i$ from another peer $j$ will result in its requests being denied by peer $j$. Hence, unless peer $i$ has already requested the same blocks from other peers in addition to peer $j$, it has to send out new requests seeking the missing blocks which can result in late arrival of those blocks. On the other hand, asking multiple members of the group for the same block increases the peer's debt (in credits), which will discredit the requesting peer (peer $i$ in this case) if it does not have enough upload bandwidth to fulfill its promises.

Based on the argument above, block requests must be sent according to the parter's reciprocation history with a minimal overlap among the requested blocks from various partners. We propose the following approach for allocating packet requests. A peer $i$ generates a packet request for each class of the missing NAL units within its sliding window and keeps them in a buffer. Next, the offered price of each requested missing block is determined based on its PID and recorded in the packet request. Then, these packet requests are sorted according to their PID's from high to low. Let $c_{ij}$ be the number of unfulfilled credits peer $i$ has already requested from another partner peer $j$. Furthermore, let $P_{ij}(c_{ij})$ be the probability that peer $j$ sends back a total of $c_{ij}$ credits. Then, for each packet request in the sorted list, assuming that it offers $\delta c_i$ credits to the receiving node, a peer $j^*$ is selected according to

$$j^* = \arg\max_{j \in G_i} P_{ij}(c_{ij} + \delta c_i). \quad (13)$$

A message is then generated and sent to $j^*$ requesting the missing packet in return for $\delta c_i$ credits and $c_{ij}$ is incremented to $c_{ij} + \delta c_i$. This process continues until a request is sent out for every missing packet. The probability $P_{ij}(c_{ij})$ can be easily estimated by peer $i$ for each of its partners based on its current action $\mathbf{a}_i$ (which is already decided by the peer) and the state transition probabilities $P_{\mathbf{a}_i}(\mathbf{s}_i, \mathbf{s}_i')$.

If a packet requested from peer $j$ is not received at time $t - \delta t$, where $t$ is the expiration time of the packet and $\delta t$ is a fixed duration set by the user, then, a new request for the packet is generated according to (13) such that $j^* \neq j$. The best value for $\delta t$ can be determined via experiment.

## 5. Experimental results

In this section we present the results of our simulations to show the effectiveness of the proposed techniques. We assume that a group consisting of $n$ nodes has already been formed. Before the streaming starts, nodes send out a BM message notifying each other about their available content. In our simulations, the size of the notification messages are ignored in the network since they are usually much smaller than data packets. However, in a real network simulation the effect of these messages shall not be ignored. In addition, we consider both lossless and lossy links connecting the peers in a group. Each reciprocation decision is valid for 33 ms. After this duration, called the reciprocation period, decisions of other nodes become available and thus transition probabilities are updated accordingly and a new reciprocation decision is made based on the updated transition probabilities. The reciprocation period can be made sufficiently large in real applications, where a typical session's length is at least a few hours, to reduce the complexity of the system. Ten states (different rate/credit points) are considered per peer. Four iterations for the value iteration of (6) is used for the case of "foresighted" MDP peers.

Initially, peers do not know each other, i.e., $P_{\mathbf{a}_i}(\mathbf{s}_i', \mathbf{s}_i) = P_{\mathbf{a}_i}(\mathbf{s}_i'', \mathbf{s}_i)$ for all $\mathbf{s}_i$, $\mathbf{s}_i'$, $\mathbf{s}_i''$ and $\mathbf{a}_i$; therefore, they equally divide their upload bandwidth to all members of the group. Then, as the peers continue to interact with each other state transition probabilities are observed and thus become more accurate and distinct. Collection of the statistics for evaluation of the system only starts after 15,000 resource reciprocation periods has elapsed. This helps the nodes to build a reciprocation history and pass the non-equilibrium initial state before the evaluation starts.

### 5.1. General file sharing experiments

In this experiment, we consider the scenario in which six nodes are connected to each other and sharing six different files. Each node is identified by a number $i$ between 1 and 6. To evaluate the performance of the system, we utilize the time-averaged fairness ratio (TAFR) defined for a peer as the ratio of data uploaded to data downloaded, averaged over the entire experiment duration [22]. Fig. 4 presents the empirical Cumulative Distribution Function (CDF) of the TAFR for regular BitTorrent and the proposed approach. The results are generated

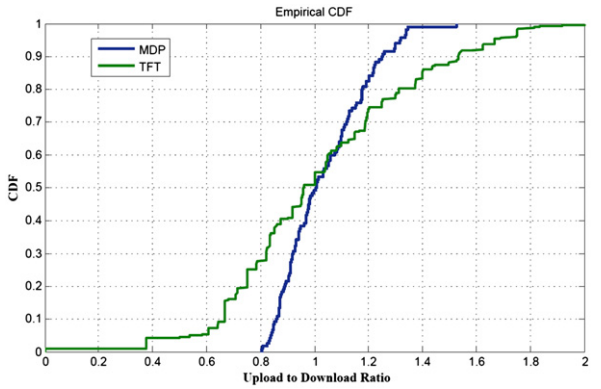


**Fig. 4.** Empirical CDF of the upload to download distribution (lossless channel).

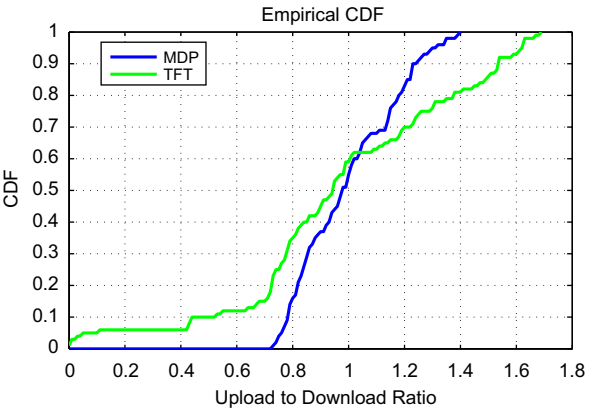


**Fig. 5.** Empirical CDF of the upload to download distribution (5% lossy channel).

from 1000 different realizations. At each realization a different seed is used per peer and its upload capacity is randomly selected from the range of $100i$ kbps to $100i + 99$ kbps, where $i$ denotes the peer's number. The BitTorrent implementation uses the concept of optimistic unchocking with a period equal to 20 reciprocation periods. In the BitTorrent approach, peers with low capacities tend to download disproportionately more data than they provide to other peers. This is attributable to the BitTorrent's optimistic unchock 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, this system ensures that data is transferred for at least 20 reciprocation periods (equal to 0.6 s). In the proposed system on the other hand, peers accurately follow each others' reciprocation behavior and will not allow an upload rate that is much higher than the receiving rate for such a long duration. As a result, the proposed system can achieve much more in terms of fairness when compared to the BitTorrent implementation.

Next we consider a scenario similar to the one shown above but with imperfect channel conditions. Specifically, the channel is considered to be lossy with a packet loss probability of 5%. The loss of packets recorded in peers' history database is taken into account when peers determine their optimal reciprocation policies. Since the packet losses are assumed to be uniform, the channel impairment does not hurt any particular policy more than the others. Fig. 5 shows the accumulative distribution of upload to download ratio. The figure is based on 20 channel loss realizations, where in each realization peers carry out 16,000 reciprocation cycles with others. As can be seen from the figure, similar to the case of lossless channel, the MDP approach yields better fairness among peers with varying upload capabilities than its TFT counterpart.

In a second experiment, we consider the groups shown in Fig. 6. In the first case (Fig. 6(a)), five peers that use the BitTorrent's tit-for-tat (TFT) policy are connected together. We refer to these peers as TFT peers. In a second case, peer number 1 is replaced by a peer that uses the proposed foresighted optimization method. We refer to this peer as the MDP peer since it uses the MDP framework (with four iterations of the value function) described in Section 4. Fig. 7 shows the average and instantaneous download rates of peer 1 for the first and second configurations (TFT and MDP cases). Optimistic unchocking is not used in this case. Fig. 8 demonstrates the same experiment only with optimistic unchocking. To illustrate how the MDP peer achieves a higher average download rate, we plotted the download rates of each case (i.e., optimistic unchocking off or on) versus time for one realization in Figs. 7(a) and 8(a). Each point on the time axis of the figure represents 50 reciprocation periods. Only in this experiment, the download rates are measured from the beginning when resource reciprocation starts. As expected, the MDP peer is able to quickly learn the behavior of the TFT peers and thus reach download rates much higher than it provides to others by just choosing the right upload rates for each of its partners. As it can be seen in Figs. 7(a) and 8(a), the learning speed and the average download to upload ratio of the MDP peer (compared to the TFT peer) is much lower when optimistic unchocking is used. The stochastic nature of optimistic unchocking violates the assumption of the MDP peer that its partners are rational and self-interested because at some instances, for a fixed duration, they choose to upload to other peers without expecting them to reciprocate. Furthermore, partly random behavior of the TFT peers causes the state transition probabilities kept by the MDP peer to be inaccurate and therefore degrades its performance.

### 5.2. Video streaming experiments

In this section, we conduct an experiment that involves video streaming instead of just generic file sharing. Five peers are considered in the group watching the Foreman CIF ($288 \times 352$) video sequence at 30 frames per s (fps). Their upload speeds are $\{240, 320, 400, 480, 560\}$ kbps. The bit stream is generated using JSVM (the scalable profile of the H.264/AVC) reference software [23]. A GOP size of 8

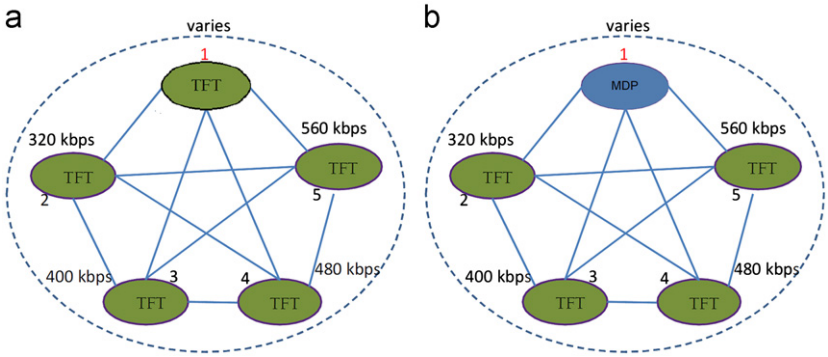


**Fig. 6.** Test configuration for mixed peer type experiment. (a) TFT peer types. (b) Mixed peer type.

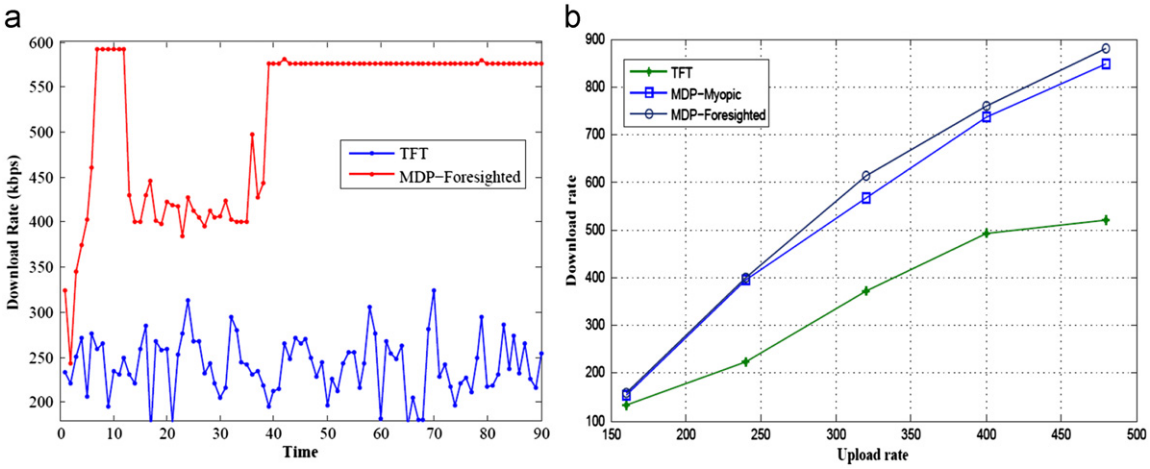


**Fig. 7.** Comparison of download rates of MDP and TFT peer; optimistic unchocking is off.

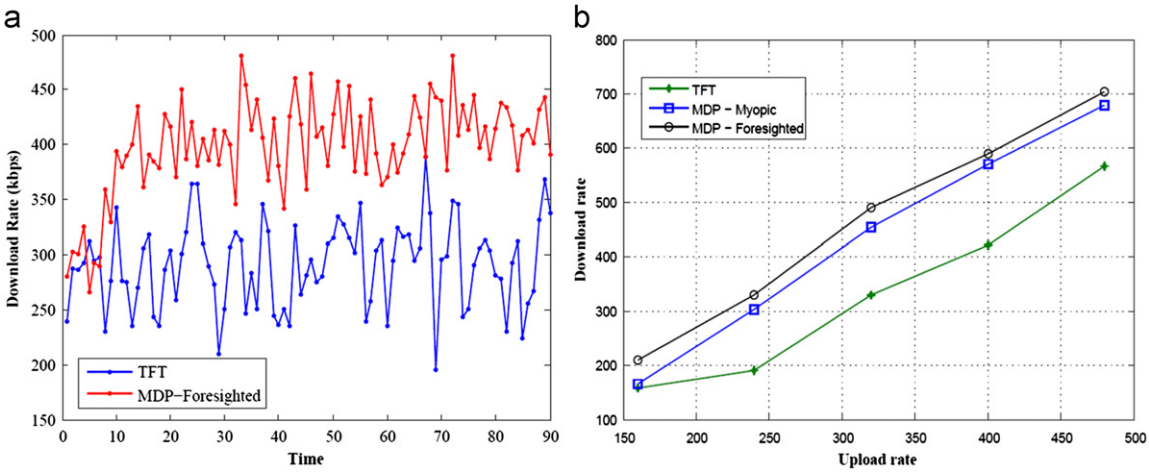


**Fig. 8.** Comparison of download rates of MDP and TFT peer; optimistic unchocking is on.

frames is considered. The sequence is encoded into two layers. One base layer, with a basis QP of 32, and one enhancement layer, with a basis QP of 24. The enhancement layer is further divided into four MGS quality layers. The sliding window considered consists of four GOPS. Each GOP, consisting of 40 NAL units is considered to be one piece. Furthermore, due to the small size of the sliding window, each NAL unit is considered to be one block in this experiment. All the NAL units within a GOP are then classified into six different PID classes utilizing the Lloyd–Max quantization of their impact on the average distortion measured in mean-squared-error (MSE). The offered per

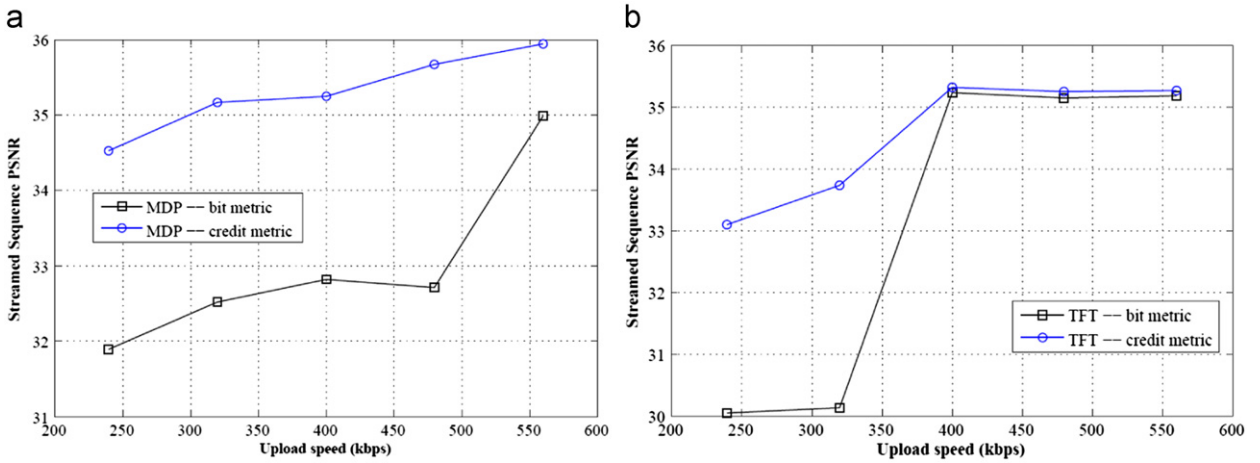


**Fig. 9.** PSNR when using bit or credit measure. (a) MDP peers. (b) TFT peers.

bit price $\rho$ for each of the six classes is manually set to {1.8,1.6,1.5,1.2,1.1,1.0} credits. If a particular block (NAL unit) that belongs to the first GOP of the sliding window (which will be displayed next) is missing, a duplicate request is sent to another peer to get the missing block. In other words, the tolerance time $\delta t$ defined in Section 4 is set to $8 \times 33 = 264$ ms. If the missing block does not arrive before its expiration time, the decoder will have to decode the sequence without that particular block. In the case that the missing block corresponds to a base-layer frame, the decoder uses the closest temporal neighbor to replace the missing picture. Since the key pictures for the entire sequence (300 frames) are coded as P pictures, their loss will cause the rest of the sequence to become undecodable. Moreover, a NAL unit is discarded by the decoder if it depends on another NAL unit, according to the structure of the bit stream, which is unavailable.

The starting frame for each of the peers is selected randomly (between 1 and 300). If the end of the sequence is reached, the playback point is rolled back to the starting frame to allow arbitrarily long sessions. The first frame of the sequence is an I frame and is assumed to be available to all peers at the beginning of the session. Even though peers have the entire sequence available for upload to others, in this experiment, they only use what they have downloaded within the duration of the sliding window to evaluate the quality of the sequence. Average PSNR results shown here are an average of 1900 reciprocations which is equivalent to 1800 frames considering 30 fps streaming.

We compare two different systems in our experiment. The first system uses the proposed MDP resource reciprocation scheme but all bits are assumed to be equally important. Thus, the amount of resources in the reciprocation game is measured in bits per second. The second system, however, uses the unequal price offering scheme of Section 3. Fig. 9(a) demonstrates the performance of each system in PSNR. As seen in the figure, all peers of the group benefit from introducing this price mechanism. This is because when a peer needs an important piece of data that has a big impact on its video's visual quality, other

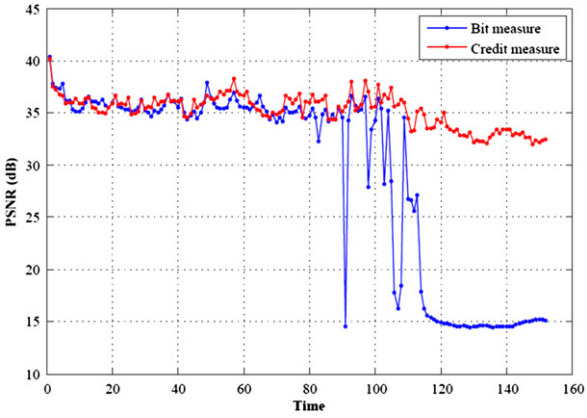


**Fig. 10.** PSNR of peer 1 per frame when using bit or credit measure.

self-interested peers are more motivated to provide the missing piece if the peer agrees to pay them back more than what it is receiving now (in terms of bits). This mechanism increases the chance of receiving important packets for all peers and therefore improves the video quality of each peer despite the fact that some peers may experience a lower average bit rate. To be able to further investigate how this mechanism helps the average performance, in Fig. 10, we plotted the PSNR per frame for a streaming session. As seen in the figure, the performance of the two systems is comparable in the beginning of the session where the sequence has low activity. However, when the camera panning starts around frame number 100 and as a result, the bit rate increases, important base-layer packet is not received on time and therefore their corresponding frames are dropped in the first system. The second system prevents this from happening by motivating its partners to give higher priority to these crucial packets.

In addition to the MDP system, the performance of the TFT system can be also improved by adding the aforementioned price mechanism. A TFT system can be envisioned in which resources are measured in credits. Thus, each peer divides its upload bandwidth equally between peers who

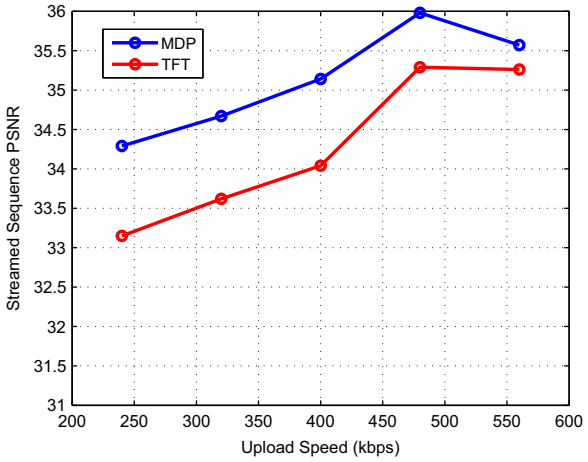


**Fig. 11.** PSNR for MDP and TFT peers (5% channel loss).

contribute the most in its download (measured in credits). In addition, the probability that a chocked peer (due to insufficient recourse contribution) is randomly unchocked in such system is assumed to be proportional to the maximum offered price for the pending packet requests made by that peer. This contrasts the original optimistic unchocking concept since the unchocking probability can be potentially different for each peer. Fig. 9(b) presents the performance of the original TFT system (bit measure) and the modified TFT system which uses the credit resource measuring scheme for each of the five peers. The average received PSNR is improved for all peers, however, the gain is significant for peers with limited upload capabilities. The credit measure TFT system increases the probability of receiving at least a base quality even when the peer has a low upload bandwidth. It should be noted that this average quality enhancement results from appropriately prioritizing the packets and hence increasing the chance of receiving them on time. Consequently, even for a significant quality improvement, the peers' average downloading rates are not necessarily increased.

Finally, we carry out a video streaming experiment involving an imperfect channel with packet loss rate of 5%. The basic experiment setting is the same as that in Section 5.2, where five peers with different upload capacities share the Foreman sequence with each other. The channel is lossy and the information loss affects both the reciprocation policy as well as the reconstructed video quality. The PSNR for both MDP and TFT algorithms are plotted. Note here the credit metric is used for both cases. As can be seen from Fig. 11, the MDP algorithm performs better in terms of PSNR than its TFT counterpart even in the case of lossy channels.

## 6. Conclusion

In this paper we considered the problem of content-aware, foresighted resource reciprocation for multimedia content streaming over P2P networks consisting of self-interested and rational peers. We have taken the video content and signal characteristics into account by proposing an artificial credit currency. The resource reciprocation process was modeled as a stochastic model and an MDP framework was introduced to determine the optimal strategies. The performance improvement by the proposed techniques over traditional approaches has been confirmed by experimental results.

## References

[1] E. Maani, A.K. Katsaggelos, A game theoretic approach to video streaming over peer-to-peer networks, in: IEEE 17th International Conference on Image Processing (ICIP), Hong Kong, September 2010, pp. 2909–2912.
[2] BitTorrent, 〈http://www.bittorrent.com〉.
[3] J. Liu, S. Rao, L. Bo, H. Zhang, Opportunities and challenges of peer-to-peer Internet video broadcast, Proceedings of the IEEE, vol. 96(1), 2008, pp. 11–24.
[4] S. Androutsellis-Theotokis, D. Spinellis, A survey of peer-to-peer content distribution technologies, ACM Computing Surveys 36 (4) (2004) 335–371.
[5] X. Zhang, J. Liu, B. Li, Y.-S. Yum, CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming, INFOCOM, Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, March, vol. 3, 2005, pp. 2102–2111.
[6] X. Jiang, Y. Dong, D. Xu, B. Bhargava, GnuStream: a P2P media streaming system prototype, Proceedings of the International Conference on Multimedia and Expo'03, July, vol. 2, 2003, pp. II-325–II-328.
[7] J. Li, PeerStreaming: A practical receiver-driven peer-to-peer media streaming system, Microsoft Research MSR-TR-2004-101, 2003.
[8] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, A.E. Mohr, Chainsaw: eliminating trees from overlay multicast, in: Peer-to-Peer Systems IV 4th International Workshop, IPTPS, Ithaca, NY, USA, February 2005.
[9] A. Legout, N. Liogkas, E. Kohler, L. Zhang, Clustering and sharing incentives in BitTorrent systems, Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, June, vol. 35(1), 2007, pp. 301–312.
[10] B. Yu, M.P. Singh, Incentive mechanisms for peer-to-peer systems, in: Agents and Peer-to-Peer Computing Second International Workshop, AP2PC, Melbourne, Australia, July 2003.
[11] C. Buragohain, D. Agrawal, S. Suri, A game theoretic framework for incentives in P2P systems, in: Proceedings of the Third International Conference on Peer-to-Peer Computing, September, 2003, pp. 48–56.
[12] J. Shneidman, D.C. Parkes, Rationality and self-interest in peer to peer networks, in: Peer-to-Peer Systems II Second International Workshop, IPTPS, Berkeley, CA, USA, February 2003.
[13] K. Lai, M. Feldman, I. Stoica, J. Chuang, Incentives for cooperation in peer-to-peer networks, in: Proceedings of the Workshop on Economics of Peer-to-Peer Systems, 2003.
[14] H. Park, M. van der Schaar, A framework for foresighted resource reciprocation in P2P networks, IEEE Transactions on Multimedia 11 (1) (2009) 101–116.
[15] H. Park, M. van der Schaar, Evolution of resource reciprocation strategies in P2P networks, IEEE Transactions on Signal Processing 58 (3) (2010) 1205–1218.
[16] J. Park, M. van der Schaar, Pricing and incentives in peer-to-peer networks, in: IEEE Proceedings of INFOCOM, San Diego, CA.
[17] Joint Draft ITU-T Rec. H. 264|ISO/IEC 14496-10/Amd. 3 Scalable Video Coding.
[18] B. Cohen, Incentives build robustness in BitTorrent, in: Workshop on Economics of Peer-to-Peer Systems, 2003.
[19] E. Maani, A.K. Katsaggelos, Optimized bit extraction using distortion modeling in the scalable extension of H.264/AVC, IEEE Transactions on Image Processing 18 (9) (2009) 2022–2029.
[20] D.P. Bertsekas, Dynamic Programming and Stochastic Control, Academic Press, 1976.
[21] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, 2006.
[22] J. Pouwelse, P. Garbacki, D. Epema, H. Sips, The BitTorrent P2P file-sharing system: measurements and analysis, in: Peer-to-Peer Systems IV 4th International Workshop, IPTPS, Ithaca, NY, USA, February 2005.
[23] J. Reichel, H. Schwarz, M. Wien, Joint Scalable Video Model 11 (JSVM 11), Joint Video Team, Doc. JVT-X202, July 2007.