

# Using Layered Video to Provide Incentives in P2P Live Streaming

Zhengye Liu<sup>†</sup>, Yanming Shen<sup>†</sup>, Shivendra S. Panwar<sup>†</sup>, Keith W. Ross<sup>‡</sup> and Yao Wang<sup>†</sup>

Dept. of Electrical and Computer Engineering<sup>†</sup>

Dept. of Computer and Information Science<sup>‡</sup>

Polytechnic University

{zliu04,yshen02}@utopia.poly.edu, {panwar,ross,yao}@poly.edu

## ABSTRACT

In this paper, we design a distributed incentive mechanism for mesh-pull P2P live streaming networks. In our system, a video is encoded into layers with lower layers having more importance. The system is heterogeneous with peers having different uplink bandwidths. We design a distributed protocol in which a peer contributing more uplink bandwidth receives more layers and consequently better video quality. Previous approaches consider single-layer video, where each peer receives the same video quality no matter how much bandwidth it contributes to the system. The simulation results show that our approach can provide differentiated video quality commensurate with a peer's contribution to other peers, and can also discourage free-riders. Furthermore, we also compare our layered approach with a multiple description coding (MDC) approach, and conclude that the layered approach is more promising, primarily due to its higher coding efficiency.

## 1. INTRODUCTION

With the widespread adoption of broadband residential access, P2P video live streaming has become a popular service in the Internet. Several streaming systems have been successfully deployed, serving tens of thousands of simultaneous users who watch channels at rates between 300 kbps to 1Mbps [10,11,16,20]. All of those systems use a mesh-pull design [7,20].

In these live streaming systems, free-riding is a potential problem, similar to what has been observed in P2P file sharing systems [5]. The reason is that the existing systems provide the same video quality to all peers, no matter what their individual upload contributions are. In a P2P streaming system, participating peers may have different upload

bandwidths. Institutional peers have high-bandwidth access, while residential peers with DSL and cable access have relatively low upload bandwidth.

An efficient incentive mechanism is critical to the performance of a P2P system. BitTorrent [3], one of the most popular P2P file downloading system, largely solves the incentive problem by adopting a tit-for-tat strategy [4]. However, designing incentive mechanisms for live video streaming is more challenging than for traditional file downloading. In file downloading, the primary performance measure is download time. In video streaming, video quality is the primary performance measure. Thus, the natural mechanism to incentivize users to upload at higher rates is to provide users with video quality that is commensurate with their upload rates.

To provide incentives in P2P live streaming, in this paper, we consider applying layered video with a tit-for-tat like strategy on mesh-pull P2P live video streaming systems. With layered coding, a video is coded into layers with nested dependency: a higher layer refines the video generated by lower layers. More received layers provide better video quality. In [7], it was observed that in a mesh-pull system, there exists bidirectional video traffic exchange between a pair of peers even in a small time scale ( $< 1$  minute), similar to BitTorrent. Based on this observation, we propose a distributed incentive strategy, where each peer measures its download rates from its neighbors, and reciprocates by providing a larger fraction of its upload rate to the neighbors from which it is downloading at higher rates. The peers evaluate their neighbors upload contributions in a distributed manner. This avoids using a centralized architecture and enhances the scalability of the system. Under this mechanism, a peer with a higher upload contribution is more likely to obtain a larger share of neighbors' upload rates, thus receiving more layers and having better video quality; a peer with a lower upload contribution is more likely to receive fewer layers, thus receiving a lower but still acceptable video quality. Importantly, a free-rider with no contribution to its neighbors is less likely to be served by its neighbors.

In this paper, after describing our incentive scheme with layered video, we carry out an extensive simulation study of the proposed incentive mechanism on a mesh-pull P2P live video streaming system. Our simulation takes into account the rate-distortion (R-D) characteristics of videos encoded by real video coders. This reflects the received video quality experienced by a user. The simulation results show that (i) our approach can provide differentiated video quality com-

\*This work is supported in part by the National Science Foundation under Grant CNS-0435228, and also in part by the New York State Center for Advanced Technology in Telecommunications (CATT).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

P2P-TV'07, August 31, 2007, Kyoto, Japan.

Copyright 2007 ACM 978-1-59593-789-6 ...\$5.00.

mensurate with a peer’s contribution to other peers; (ii) free-riding can be prevented due to the unacceptable video quality that free-riders receive. We also compare layered video with multiple description coding (MDC) [17] and show that the layered approach achieves better video quality due to its higher coding efficiency.

## 2. RELATED WORK

There have been several studies of incentives in P2P file downloading. The studies of BitTorrent [2, 8, 13] show that the tit-for-tat strategy can largely prevent free-riding. To date, only a few strategies have been proposed to provide incentives in P2P video live streaming [6, 15]. In [6], Habib et al. proposed a rank-based peer-selection mechanism for a P2P media streaming system. In this mechanism, the contribution of a user is represented by a score, which is then mapped to the rank of this user among all users in the system. A peer with higher score has more flexibility in peer selection, while a free-rider has limited flexibility in peer selection. Similarly, in [15], a payment-based incentive mechanism is proposed. Using this approach, peers earn points by forwarding data to others and compete with each other for good parents in a first-price auction like procedure. Both of these approaches use reputation or micro-payment based mechanisms. However, typically, such solutions use a centralized architecture for coordination that can hinder the scalability of a P2P streaming system. In our previous work [9], we considered using MDC with a tit-for-tat like strategy in a mesh-pull video streaming system. With MDC, each substream has the same importance, which greatly simplifies the system design. However, MDC involves substantial redundancy: for the same reconstructed video quality, the total bit rate of all received descriptions is typically much higher than that of all received useful layers.

## 3. SYSTEM ARCHITECTURE AND DESIGN

### 3.1 MDC vs. Layered Coding

In a previous paper [9], we considered using MDC video to provide incentives in P2P streaming. With MDC, a video is encoded into several descriptions with equal importance. When a video is encoded into  $M$  descriptions, the combination of any  $m \leq M$  different descriptions are decodable, with more descriptions resulting in a better video quality. Although MDC simplifies the system design, existing MDC coders incur significant redundancy. With the same transmission bit rates, the video quality of MDC is much lower than that of a single layer coded video or a layered video. For these reasons we explore layered coding in this paper. Unlike MDC, with layered coding, a video is encoded into several layers with nested dependence. A higher layer can be decoded only if all the lower layers are available. Generally, layered coding does not lose much video coding efficiency compared with single layer coding. Our simulation results in Sec. 4 show that layered coding can indeed have much better performance than MDC. But layered coding in P2P streaming also has its challenges:

- Unequal importance of different layers: due to the nested dependence of layers, scheduling algorithms need to assign higher priorities to the lower layers than to the higher layers.

- Layer availability among neighbors: In our incentive mechanism, a peer with a higher contribution should receive more layers than a peer with a lower contribution. It is possible that no neighbor of a peer holds the higher layers desired. To obtain the layers commensurate with its upload contribution, a high-bandwidth peer should be able to locate other high-bandwidth peers that have the higher layers.

### 3.2 A Mesh-Pull Live Video Streaming System with a Tit-for-Tat Strategy

In a mesh-pull delivery architecture for live video streaming, a video is divided into media chunks, and the chunks are made available at an origin server. When a peer wants to view the video, it obtains a list of peers currently watching the video and establishes partnership with several peers. Peers exchange chunk availability information with their neighbors and request chunks that they need.

In our proposed system, the origin server encodes a video into  $L$  layers. Each layer is further divided into Layer Chunks (LCs) of  $\Delta$  seconds. Partners exchange the LCs with each other. Each peer measures its download rates from its neighbors. A peer reciprocates to its neighbors by providing a larger fraction of its upload rate to the neighbors from which it is downloading at the highest rates. In this manner, a peer with higher upload contribution is likely to be rewarded with more LCs, and hence more layers and better quality.

Our proposed strategy applies both supplier side scheduling and receiver side scheduling. As a supplier, a peer receives LC requests from multiple neighbors. The peer must determine which request should be served first and how to allocate its available uplink bandwidth to its neighbors. As a receiver, a peer must prioritize its requests for its missing LCs. And for each LC to be requested, it must determine to which partner peer it sends the request. In this paper, to simplify the discussion, we assume there is no downlink bandwidth constraint. (A downlink bandwidth constraint can be easily incorporated, and does not change the findings in this paper.)

### 3.3 Supplier Side Scheduler

In our design, a peer will upload more to the neighbors from which it downloads more. To this end, a supplier maintains a different request queue for each receiver. For a particular receiver, the queue is first-in-first-out, where the supplier serves the requests in the order that the requests were received. The supplier transmits one requested LC to one receiver at one time. The supplier determines which receiver should be served depending on the receiver’s contribution to the supplier. At any one time, the supplier randomly selects a receiver to serve. Let  $p_{n,k}$  denote the probability that peer  $n$  selects the receiver  $k$ .  $p_{n,k}$  is determined as follows:

$$p_{n,k} = \frac{I_{n,k} d_{n,k}^{\gamma}}{\sum_{i \in \mathcal{K}_n} I_{n,i} d_{n,i}^{\gamma}}. \quad (1)$$

where  $\mathcal{K}_n$  denotes the set including all neighbors of peer  $n$ , and  $d_{n,i}$  is the estimated download rate of supplier  $n$  from receiver  $i$ .  $d_{n,i}$  can be obtained based on the number of LCs delivered from peer  $i$  to peer  $n$  during the previous time period.  $I_{n,i}$  equals 0 if the request queue of receiver  $i$  is empty, and equals 1 otherwise.  $\gamma$  is a parameter that controls the sensitivity of peer  $n$  to peer  $k$ ’s contribution. Therefore, a supplier serves its neighbors in a weighted fashion, and

a receiver that uploads more to the supplier has a higher probability of being served, thus consuming a larger share of the supplier’s uplink bandwidth. In Sec 3.5, we provide a refinement of the supplier side scheduler.

### 3.4 Receiver Side Scheduler

At any given time, a receiver will have buffered LCs that are to be displayed in the future. Figure 1 shows an example for the LCs buffered at a particular receiver, where the X’s denote buffered LCs. In this example, there are 8 LCs buffered: 3 for the next chunk time, 2 for the second chunk time, and so on. Other LCs have been requested but have not yet arrived. These are shown with dashes in Fig. 1. In this example, there are three such requested LCs. There are also LCs available in the neighbors, but have not been requested. These LCs are left blank in Fig. 1. Additionally, a non-buffered LC may not be available from any of the neighboring peers. These are shown by circles in Fig. 1. These buffered LCs, requested LCs, available LCs, and not-available LCs constitute the current *buffer state* of the receiver.

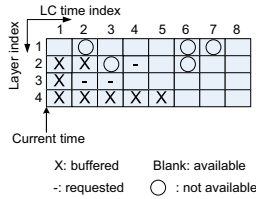


Figure 1: Buffer state at a given time.

In our design, peers request LCs at the beginnings of rounds. At the beginning of each round, the peer will have a buffer state, as described above. Given this buffer state, the peer must decide which LCs to request. The peer may request LCs from the current time up until a window of  $B$  chunk times into the future. Figure 1 uses a window of  $B = 8$ .

Normally, at the beginning of each round, there are several available-not-requested LCs. The problem is: which LC should be requested first and from which neighboring peer? We propose a simple heuristic to solve this scheduling problem. A more sophisticated scheduling algorithm, such as the algorithm in [19], can be adapted to this framework.

In our algorithm, we assign a score to each available-not-requested LC. Chunks with lower scores get requested with higher priority. Three factors of a LC are jointly considered to calculate the score, namely, (i) layer index, (ii) playback deadline and (iii) rarity. With layered coding, since a layer can be decoded only if all the lower layers have been received, lower-layer LCs should be given higher priority. Also, a LC in danger of being delayed beyond the playback deadline should be more urgently requested. Additionally, as in BitTorrent, rare chunks should be given higher priority. Generally, for a peer  $n$ , the score  $S_{l,t}^n$  of a LC with layer index  $l$  and playback deadline  $t$  can be expressed as:

$$S_{l,t}^n = G(l, t, \lambda), \quad (2)$$

where  $\lambda$  is the number of duplicates of this LC available in the neighbors of peer  $n$ .  $G(\cdot)$  is a monotonously increasing function with  $l$ ,  $t$  and  $\lambda$ . Specifically, in this work, we apply a weighted linear combination of these three factors, mathematically expressed as:

$$S_{l,t}^n = w_1 \frac{l}{L} + w_2 \frac{t - t_0}{B} + w_3 \frac{\lambda}{K}, \quad (3)$$

where  $t_0$  is the current time,  $L$  is the total number of layers,  $B$  is the buffer size in terms of LC times, and  $K$  is the number of neighbors of peer  $n$ .  $w_1$ ,  $w_2$  and  $w_3$  are the weights of the three factors. Our simulation studies indicate that layer index has more impact on performance and therefore should have a relatively larger weight.

A receiver requests the available-not-requested LCs from the lowest score to the highest score. Given a LC, a receiver  $n$  can estimate the time  $\tau_{n,k}$  that this LC will be received from its neighbor peer  $k$  using the following equation:

$$\tau_{n,k} = t_0 + C_{n,k} \frac{r\Delta}{d_{n,k}}, \quad (4)$$

where  $C_{n,k}$  is the number of outstanding requests from  $n$  to  $k$ ,  $r$  is the bit rate of one layer, and  $d_{n,k}$  is the estimated download rate of peer  $n$  from peer  $k$ . In our heuristic, a peer will send a LC request to the neighbor that can deliver this LC at the earliest time. Therefore, if  $\tau_{n,k}$  is the minimum, and if it is less than this LC’s playback deadline, then peer  $n$  will send the request to peer  $k$ . If this LC cannot meet its playback deadline, it will not be requested, and the receiver will request the LC with next higher score.

### 3.5 Download Rate Probing

In the receiver side scheduling, peer  $n$  schedules requests to peer  $k$  based on the current estimated  $d_{n,k}$ . In particular, our algorithm (as described so far) does not request a LC that is expected to arrive after its deadline (see (4)). However,  $d_{n,k}$  may not match the potential rate that peer  $n$  can receive from peer  $k$ . If a receiver requests too conservatively, it may not fully utilize the potential bandwidth that its supplier has available to allocate to it. On the other hand, if this receiver requests too aggressively, many of its requested LCs may not be served before their deadlines. In this case, the LC requests in a later request round may be blocked by the earlier requests, no matter what their priorities are. In this section, we propose a refinement algorithm over the basic supplier/receiver scheduler algorithm.

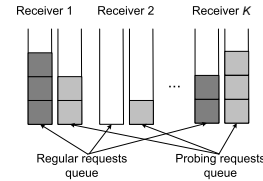


Figure 2: Request queue structure at a supplier.

With this design, in addition to sending the *regular requests* based on the receiver side scheduling described in Sec. 3.4, each receiver also sends some *probing requests* to its neighbors. A probing request has a lower service priority than a regular request. The probing requests try to utilize the available bandwidth to refine the received video quality, and if they are not served, it does not have much impact on video quality. As shown in Fig. 2, in addition to maintaining one request queue as described in Sec. 3.3, a supplier maintains another probing request queue for each receiver. In both of these queues, the requests are served in a first-in-first-out manner. For each receiver, the requests in the probing requests queue can be served only if the regular requests queue is empty. Therefore, if receiver 2 in Fig. 2 is selected for service, the requests in the probing requests queue will be served. Note that the regular requests are scheduled based on the current estimated download rate

$d_{n,k}$ . For receiver  $n$ , if the potential rate is higher than  $d_{n,k}$ , supplier  $k$  serves the probing requests of receiver  $n$  after finishing serving all the regular requests, so that the potential upload bandwidth of supplier  $k$  will be probed. If the potential rate is less than  $d_{n,k}$ , the probing requests will not be served, but they will not block the regular requests. The supplier will discard any request that exceeds the deadline of that receiver. These probing requests can be piggybacked in the regular request messages, so that the probing requests do not introduce much additional overhead.

### 3.6 Partner Selection

Like BitTorrent, in order to locate a better partner with higher uplink bandwidth and more content, in our scheme a peer periodically replaces the partner with the least contribution with a new peer. To establish a stable relationship, it is desirable for the peer to locate a partner with similar uplink bandwidth. If the uplink bandwidth of the partner is too low, the peer can only receive the lowest layers from the partner. Thus, this partner may not hold the peer's missing layers. On the other hand, if the uplink bandwidth of the partner is too high, it is likely that other neighbors of the partner have higher uplink bandwidth, so that the peer will be dropped by the partner after several rounds.

To locate a suitable partner, one simple approach is to randomly select a peer from the active peers in the system. To accelerate this process, a peer can select a partner from a pre-screened set of candidates. For example, a peer can choose a partner with the same ISP or with the same type of access. These peers are expected to have similar uplink bandwidth. Another approach to locate a candidate is based on peer's buffer maps [18]. When two peers intend to establish partnership, they exchange their buffer maps before establishing partnership. Since a peer with a higher contribution is supposed to receive more layers, and vice versa, the uplink bandwidth contribution of a peer can be evaluated based on its buffer map.

## 4. PERFORMANCE STUDIES

We compare four schemes in our simulation: (i) Single layer video without incentives (Single-Layer): a video is encoded by single layer video coding; (ii) Layered video without incentives (Layered): a video is encoded into multiple layers, however, the upload contribution is ignored and each peer is treated equally; (iii) MDC with incentives (MDC-Incent) [9]: a video is encoded into multiple descriptions, and the received video quality of a peer depends on its upload contribution; (iv) Layered video with incentives (Layered-Incent): our proposed scheme.

### 4.1 System Setup

In our simulation, there are 1000 peers in the overlay, with each peer being either an Ethernet peer with 1000 kbps uplink bandwidth, a cable peer with 300 kbps uplink bandwidth, or a free-rider. We fix the ratio between Ethernet peers and cable peers as 3:7, and change the percentage of free-riders in the system. Note that without free-riders, the average upload bandwidth is 510 kbps. The video is initially stored at an origin server and the origin server has an uplink bandwidth of 6 Mbps. The origin server will send out chunks from all layers in round-robin order to its requesting peers. In this way, the chunks from higher layers can be dispersed into the system. Since low bandwidth peers do not

need chunks from higher layers, the system will reconfigure itself to ensure that the neighbors of the origin server are all high bandwidth peers. The request round is 1 second and the buffer length is 30 seconds. Each peer estimates a neighbor's contribution based on the number of chunks received from it in the previous 60 seconds. Initially, each peer has 14 randomly chosen neighbors. Peer  $k$  will accept the partnership request from peer  $n$  if peer  $k$  has less than 18 neighbors; otherwise peer  $k$  will reject the request. In our simulation, if peer  $k$  has more than 14 neighbors, when it drops the neighbor with the least contribution, it will not search for a new neighbor. When peer  $k$  and peer  $n$  newly establish a partnership, they assume the bandwidth allocated from each other is 20 kbps. In the systems with incentives, during every 30 second period, a peer replaces the neighbor with the least contribution by a randomly chosen peer. We set  $w_1 = 10.0$ ,  $w_2 = 1.0$ , and  $w_3 = 2.0$ . We examine  $\gamma = 0.5$  and  $\gamma = 1.0$  for Layered-Incent.

To investigate the received video quality experienced by users, our simulation takes into account the R-D characteristics of an encoded video. We apply the most advanced scalable video coder (SVC), which is under standardization by the MPEG AVC/ITU-T group [14], on a real video sequence. We choose the "Foreman" video sequence in CIF (352x288) resolution with a frame rate of 30 frame/sec. For layered video coding, we encode this sequence into a fine-granular scalable (FGS) bit stream at a base layer rate of 50 kbps. Each group of pictures (GOP) has a duration of 1 second. The output bits from each GOP are converted to 20 layers. The bit rate of each layer is 50 kbps. Each layer is further divided into 1 second LCs.

For the MDC-Incent scheme, we apply multiple description FEC (MD-FEC) [12, 17] scheme on the above FGS encoded bit stream to generate 20 descriptions. The bit rate of each description is 50 kbps. Note that MD-FEC is a popular scheme generating multiple ( $>2$ ) descriptions from any scalable bit stream. The optimal layer partition [12] is applied to adapt to the description loss distribution. In our simulation, we only consider the description loss distributions of Ethernet peers and cable peers that have contributed to the system and ignore the free-riders.

For the single layer coding scheme, we code this "Foreman" sequence into a single layer bit stream with bit rate 400 kbps using the H.264 codec JM9.6 [1]. The GOP length is again one second. We divide each GOP into 8 chunks, with the first chunk containing the first few frames and so on. Thus, the chunk size is kept the same in all schemes. We pre-computed the operational rate-distortion function for "Foreman" based on SVC and H.264 respectively.

### 4.2 Performance Metrics

Three metrics are adopted for performance comparison, namely: (i) Useful rate received ( $R$ ): Due to the nested dependence of layers, a LC from a high layer is useful only if all its lower layers are received. To calculate the useful rate received, we only count the decodable LCs; with MD-FEC, only received source bits after MD-FEC decoding are counted in the useful rate received; with single layer coding, the chunks are naturally dependent within a GOP. When one chunk is lost, the error induced will be propagated to frames packed in the following chunks. We assume frame-freeze is adopted for error concealment, so that if one chunk is lost, its following chunks within a GOP become useless.

$R$  represents the rate averaged over a particular video session or averaged over several video sessions; (ii) Discontinuity ratio ( $\alpha$ ):  $\alpha$  is defined as the percentage of times that a video is undecodable and unplayable. For the Layered, Layered-Incent and MDC-Incent schemes, if the base layer is decoded, the video is decodable and playable; for single layer, if one chunk is lost, this chunk and its following chunks within the GOP are considered undecodable; (iii) Average PSNR ( $Q$ ):  $Q$  represents the received video quality. We calculate the PSNR  $q$  for each decoded frame, and average frame PSNR over all decoded frames.

## 4.3 Simulation Results

### 4.3.1 Differentiated service

In this first simulation, we fix the percentage of free-riders as 5%. The Corresponding percentage of Ethernet and cable peers are 28.5% and 66.5%, respectively.

Figure 3 shows the cumulative distribution of average received *useful* rates for different classes of peers. The rate  $R$  here is averaged over the entire simulation time for a particular peer. We also indicate  $(Q, \alpha)$  for each class of peers under each scheme in the figures. Figure 3(a) plots the CDF of Ethernet peers for the four schemes: *Single*, *Layered*, *MDC-Incent* and *Layered-Incent*. It shows that the Ethernet peers with Layered-Incent scheme have much higher received useful rates than with other three schemes. Accordingly, their received video qualities are also higher. In the case of  $\gamma = 0.5$ , most Ethernet peers receive a useful bit rate higher than 600 kbps, while in the case of  $\gamma = 1.0$ , the rate is higher than 800 kbps. For the MDC-Incent scheme, the optimal layer partition is designed to optimize the overall system performance. The optimal layer partition gives more protection to the peers with less received descriptions, by introducing more redundancy bits. Hence, given a total transmission rate, less source bits can be transmitted. Although an Ethernet peer can receive more descriptions than a cable peer [9], the received source rate after MD-FEC decoding is low. The received useful rates with MDC-Incent and the corresponding video quality  $Q$  are much lower than those of Layered-Incent ( $\gamma = 0.5$ ). For the schemes without incentives, an Ethernet peer is treated the same as a cable peer or a free-rider and receives the average rate of the whole system. Note that for the single layer case, just about every peer receives the video at the encoded rate of 400kbps.

For the same reason, in Fig. 3(b), a cable peer receives a lower rate than Ethernet peers with Layered-Incent. Note that for the cable peers, the received video quality is acceptable with continuous playback. Similarly, with MDC-Incent, a cable peer also receives a lower source rate than an Ethernet peer. The received video quality with MDC-Incent is worse than that with Layered-Incent.

Figure 3(c) shows the case of free-riders. With Layered-Incent or MDC-Incent, the receive rate of a free-rider is low. More importantly, the discontinuity ratio for free-rider is high (Layered-Incent: 13% and 20%, MDC-Incent: 76%), which is unacceptable for video service. This virtually prevents free-riding. With Single-Layer and Layered, a free-rider can receive the same video quality as an Ethernet peer.

### 4.3.2 Robustness to free-riding

In addition to providing differentiated services and preventing free-riders, it is also desirable for a system to be

robust to free-riding. The robustness to free-riding indicates that received video quality of peers that are willing to contribute upload bandwidth should not be affected much by free-riders. In this second simulation, we vary the percentage of free-riders in the system from 5% to 30%, and examine the system performance of different schemes. We only show the results of  $\gamma = 0.5$  for Layered-Incent. All three performance metrics are compared.  $R$  represents the average received useful rate for a particular class of peers.

Figure 4 shows the received video quality of different classes of peers under different schemes when free-riding occurs. Figures 4(a)-(c) show the case of Ethernet peers. As shown in Fig. 4(a), with Layered-Incent, the Ethernet peers receive much higher rates than the other three schemes. The received rates with Layered-Incent are almost unchanged as the percentage of free-riders increases. This indicates free-riding does not have much impact on the received video quality of the Ethernet peers. This is further verified in Fig. 4(b) and Fig. 4(c). Both the discontinuity ratio and average PSNR are constant.

Similarly, with MDC-Incent, free-riding does not affect the received useful rate and video quality. However, due to redundancy, the received useful rate is much lower than that with Layered-Incent. Accordingly, the received video quality is also much lower. The gap in the average PSNRs is about 2 dB.

In contrast, the Single-Layer scheme is seriously affected by free-riding. As shown in Fig. 4(a), the average received rate drops. Without awareness of peer contributions, all peers share the total available bandwidth with the same priority. With more free-riders, the total available upload bandwidth decreases, while the total demand stays the same. Thus, the bandwidth allocated to each peer is decreasing. As shown in Fig. 4(b) and Fig. 4(c), the received video quality drops dramatically. With Single-Layer, the video rate is fixed. Once a peer's received rate is less than the video rate, the video quality degrades severely.

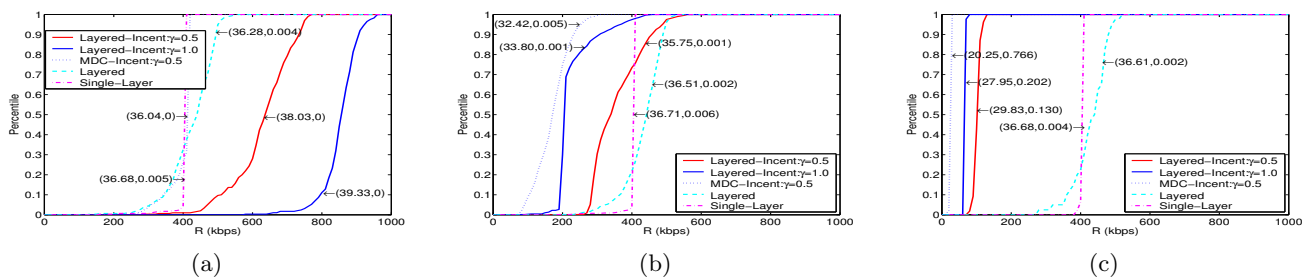
Layered scheme can mitigate the effect of free-riding to a certain degree. As shown in Fig. 4(a), similar to that with Single-Layer, the received rate with Layered approach is decreasing. However, unlike Single-Layer, the Layered scheme can adapt the received video quality to the available rate. When the received rate is decreasing, the video quality degrades gracefully, which is demonstrated in Fig. 4(b) and Fig. 4(c). However, although the video quality degradation is less severe than that with Single-Layer, the average PSNR  $Q$  decreases more than 2 dB. It also should be noticed that the Layered scheme does not prevent free-riders. As shown in Fig. 4(d)-(f), similar observations can be made for cable peers.

## 5. CONCLUSION

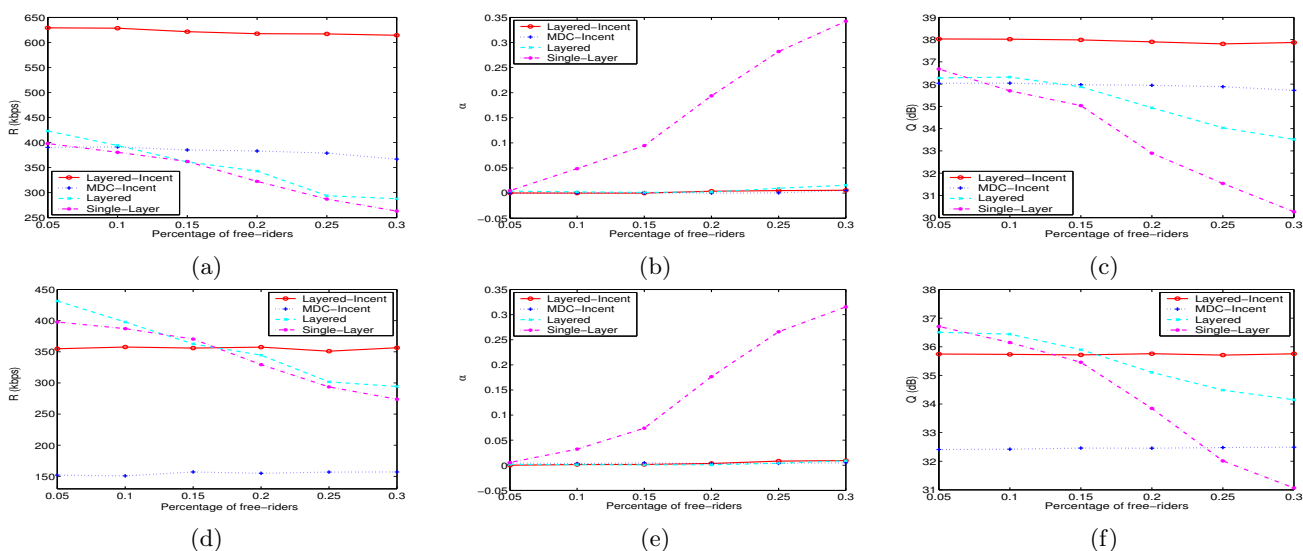
In this paper we showed that using layered video with a tit-for-tat strategy can provide incentives for peers to contribute uplink bandwidth. Moreover, the approach discourages free-riding. The framework presented here opens a number of important research questions.

## 6. REFERENCES

- [1] A. Tourapis, K. Sühring and G. Sullivan. Revised H.264/MPEG-4 AVC reference software manual. *Joint Video Team, Doc. JVT-Q042*, October 2005.



**Figure 3: Cumulative distribution of received useful rate for all peers during the simulation period under different schemes. The percentage of free-rider is 5%. Average  $(Q, \alpha)$  for each class of peers are indicated. (a) Ethernet peer; (b) cable peer; (c) free-rider.**



**Figure 4: Robustness of schemes to free-riding for different users under different performance metrics. (a)-(c) Ethernet peers; (d)-(f) cable peers.**

[2] A. Bharambe, C. Herley, and V. N. Padmanabhan. Analyzing and improving a bittorrent network's performance mechanisms. In *Proceedings of INFOCOM 2006*, April 2006.

[3] BitTorrent. <http://www.bittorrent.com/>.

[4] B. Cohen. Incentives build robustness in BitTorrent. In *1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[5] P. Colle, K. Leyton-Brown, and I. Mironov. Incentives for sharing in peer-to-peer networks. In *ACM Conference on Electronic Commerce 2001*, 2001.

[6] A. Habib and J. Chuang. Incentive mechanism for peer-to-peer media streaming. In *Proceedings of International Workshop on Quality of Service (IWQoS) 2004*, 2004.

[7] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. A measurement study of a large-Scale P2P IPTV system. In *IEEE Trans. on Multimedia*, to appear.

[8] S. Jun and M. Ahamad. Incentives in BitTorrent induce free riding. In *Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, 2005.

[9] Z. Liu, Y. Shen, S. Panwar, K. Ross, and Y. Wang. P2P video live streaming with MDC: Providing incentives for redistribution. In *ICME 2007*, Beijing, 2007.

[10] PPLive. <http://www.pplive.com/>.

[11] PPStream. <http://www.ppstream.com/>.

[12] R. Puri and K. Ramchandran. Multiple description source coding through forward error correction codes. In *33rd Asilomar Conf. Signals, Systems and Computers*, Oct. 1999.

[13] D. Qiu and R. Srikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In *Proceedings of SIGCOMM 2004*, 2004.

[14] H. Schwarz, D. Marpe, and T. Wiegand. Joint scalable video model (JSVM) 2. *Joint Video Team, Doc. JVT-O202*, April 2005.

[15] G. Tan and S. A. Jarvis. A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast. In *Proceedings of International Workshop on Quality of Service (IWQoS) 2006*, June 2006.

[16] UUsee. <http://www.uusee.com/>.

[17] Y. Wang, A. R. Reibman, and S. Lin. Multiple description coding for video delivery. *Proceeding of the IEEE*, pages 57–70, Jan. 2005.

[18] X. Hei and Y. Liu and K.W. Ross. Inferring network-wide quality in P2P live streaming systems. *submitted*.

[19] M. Zhang, C. Chen, Y. Xiong, Q. Zhang, and S. Yang. Optimizing the throughput of data-driven based streaming in heterogeneous overlay network. In *Proceedings of ACM Multimedia Modeling 2007*, January 2007.

[20] X. Zhang, J. Liu, B. Li, and P. Yum. DOnet: A data-driven overlay network for efficient live media streaming. In *Proc. of IEEE INFOCOM*, 2005.