

Issues in Offering Live P2P Streaming Service to Residential Users

Nazanin Magharei, †Yang Guo, and Reza Rejaie

Department of Computer and Information Science
University of Oregon

{nazanin,reza}@cs.uoregon.edu

†Corporate Research
Thomson Inc.

Yang.Guo@thomson.net

Abstract—Peer-to-Peer (P2P) streaming mechanism is a promising approach to offer one-to-many distribution of live streaming content to large number of residential users across the Internet. While a few P2P streaming mechanisms have been recently proposed, the behavior of these mechanisms in sessions where a large fraction of users are residential peers have not been examined. Several characteristics of residential users could affect the behavior of P2P streaming sessions including (i) the heterogeneity of access link bandwidth, (ii) the limitation of available resources due to asymmetric nature of access link bandwidth (for DSL and cable modem), and (iii) the participation of free-loaders.

In this paper, we examine the effect of these three characteristics of residential users on the performance of P2P streaming mechanisms using ns simulations. We show that these characteristics could adversely affect the performance of P2P streaming mechanisms as follows. First, the delivered quality to individual users is not correlated with their contributed resource to the system. Second, the presence of free-loaders could dramatically degrade the delivered quality to participating peers even when the amount of available resources in the system are adequate. Our findings suggest that P2P streaming has to address these issues in order to be successfully deployed among residential users.

I. INTRODUCTION

Traditionally, the client-server service model is used to provide live streaming service. A client sends a request to a server, which then streams the content to the client if the server has enough resources to serve the client's request and there is enough bandwidth along the path between the server and the client. However, this approach has a limited scalability since the available resources at the server (namely storage, CPU and bandwidth) should proportionally increase with the number of concurrent users. Another approach to offer one-to-many live streaming to a large number of users is to use IP multicast. However, the limited deployment of IP multicast limits the availability and scope of this service to interested users.

These limitations have motivated a new paradigm for live streaming based on peer-to-peer (P2P) overlays, generally being referred to as *P2P streaming*. In P2P streaming, participating users (or peers) form an overlay and actively contribute their resources (e.g., outgoing bandwidth and storage space) by forwarding the received content to other peers. The aggregate available resources in this approach organically grows with the user population and can potentially scale to any number of participating peers.

Mesh-based P2P streaming is a class of P2P streaming where participating peers form a randomly connected mesh

and employ a swarm-like content delivery mechanism similar to file-swarming mechanism such as *BitTorrent*[1].

File-swarming can effectively scale with the number of participating peers since the entire content is available at source, and there is no timing constraint for delivery of individual packets.

In live P2P streaming, the content is progressively generated by the source and should be delivered to individual peers in a timely fashion. To achieve this goal, participating peers delay their playout time with respect to source. This provides a window of new content at source that can be used for swarming. As the playout time progresses, the window of new content slides forward and the new content is swarmed among the participating peers and should be delivered before its playout time. Using larger window improves the available content and thus the effectiveness of the swarming content delivery at the cost of higher startup latency and more buffering at each peer. The key challenge is to provide a sufficiently large window of content for effective swarm-like content delivery while limiting the delay between source and participating peers as well as ensuring in-time delivery of individual packets.

Recent studies have proposed several P2P streaming mechanisms and examined some aspects of their performance through simulations or limited experiments [2–5]. These studies have often examined the performance of their proposed mechanisms in a scenario where the aggregate contributed bandwidth by all peers is greater or equal to the required system capacity (i.e., system has sufficient resource). Such a resource-rich scenarios are feasible especially when a good fraction of participating peers have high bandwidth connectivity (e.g., university users). However, in practice a large portion of participating peers are often residential users with asymmetric and heterogeneous bandwidth connectivity (i.e., users with DSL and cable modem connections). This implies that in such a setting the aggregate resources among peers may not be sufficient to serve the streaming content with the desired rate (and thus desired quality) to all participating peers. An extreme degree of heterogeneity occurs when some users may not be able (or willing) to contribute any resources and thus become *free-loaders*. These characteristics of sessions with a large fraction of residential users raise the following key question: *How do a mesh-based P2P streaming mechanism behave in an environment with either insufficient aggregate bandwidth and/or heterogeneous access link bandwidth among*

peers? More specifically, how is the delivered quality to individual peers affected as the aggregate resource decrease? How similar (or different) is such an effect across peers with different access link bandwidth? Whether and how the fraction of free-riders affect the overall performance of the system as well as delivered quality to individual peers?

In this short paper, we attempt to answer the above key questions. Toward this end, we use ns simulations to investigate the impact of the following three factors on the performance of live mesh-based P2P streaming applications: (i) aggregate resources due to asymmetric access link bandwidth, (ii) heterogeneous access link bandwidth, and (iii) percentage of free-loaders. We use *PRIME* [6] as a representative mesh-based P2P streaming mechanism. Our main findings can be summarized as follows: (i) As the aggregate resources decrease, the median delivered quality among participating peers drops proportionally. However, the impact on the delivered quality to individual peers is neither uniform nor does it depend on their contributed outgoing bandwidth. (ii) Changing the degree of heterogeneity of access link bandwidth among participating peers does not affect system performance as long as aggregate resources remain intact. (iii) the presence of free-loaders affects the connectivity of the overlay and could degrade the delivered quality to a portion of peers even if the amount of aggregate resources remain the same. We believe that our findings illustrate some potential roadblocks in employing P2P streaming services to residential users. This in turn implies that the existing P2P streaming schemes need to address these issues in order to accommodate sessions with heavily dominated residential users. While some of our findings might depend on the specific protocol used in our evaluations, we believe that some variants of the reported behavior must exist in other P2P streaming mechanism.

The rest of this paper is organized as follows. Section II describes an overview of *PRIME*, a mesh-based P2P streaming scheme. In Section III, we present our simulation setting as well as the effects of the three key factors on the performance of *PRIME*. Finally, Section IV concludes the paper.

II. PRIME: MESH-BASED P2P STREAMING SERVICE

A mesh-based P2P streaming system consists of two major components: (i) **overlay construction mechanism**, which organizes participating peers into an overlay; and (ii) **content delivery mechanism**, which determines the delivery of content from the source to individual peers through the overlay. In the following subsections, we describe these components in *PRIME*. Further details on *PRIME* description and performance evaluation can be found in [6].

A. Overlay Construction

In *PRIME*, participating peers form a randomly connected overlay. The mesh is a directed graph. Each peer maintains a certain number of parent node from which it retrieves the content, and a certain number of child nodes to whom it delivers the content. For each peer, we denote the number of parents and child nodes as peer's *incoming* and *outgoing*

degree, respectively. To effectively utilize access link bandwidth of participating peers, the incoming and outgoing degree of each peer is set proportional to its available incoming bandwidth (b_{down}) and outgoing bandwidth (b_{up}) [6]. The ratio of incoming (or outgoing) bandwidth to incoming (or outgoing) degree represents the average bandwidth of each connection which is called bandwidth-per-flow or *bwpf*. *bwpf* is a configuration parameter that is selected a priori and known by individual peers. Specifically, the incoming and outgoing degree of a peer is set to be $\lfloor b_{down}/bwpf \rfloor$ and $\lfloor b_{up}/bwpf \rfloor$, respectively. In our earlier work [6], we showed that for any given scenario there is a sweet range of peer degrees where system behaves properly.

To join the session, the new peer contacts the bootstrapping node to learn about other participating peers upon arrival. The bootstrapping node returns a list of randomly selected peers that can potentially serve as parent nodes. The new peer contacts these potential parent nodes to determine whether they are able to accommodate new child peer (*i.e.*, their current outgoing degree is less than their target outgoing degree). If a new peer can not identify a sufficient number of parents from the reported list (*i.e.*, does not reach its target incoming degree), it will contact the bootstrapping node again to obtain another list. Meanwhile, the new peer is registered with the bootstrapping node and can be selected as the parent node by other peers.

B. Content Delivery

PRIME employs a swarm-like content delivery mechanism similar to *BitTorrent*. The main advantages of the swarming content delivery are its ability to effectively utilize the outgoing bandwidth of participating peers and its robustness against the dynamics of peer participations (or *churn*).

The swarm-like content delivery incorporates push content reporting with pull content requesting. As a parent, each peer periodically reports its newly received packets to its child peers. As a child, each peer periodically requests a subset of required packets from each parent based on the reported available packet at each parent and its available bandwidth. The requested packets from each parent are determined by a *packet scheduling* algorithm. Each parent peer delivers the requested packets by each child peer through a congestion controlled connection (such as TCP or RAP).

To accommodate the bandwidth heterogeneity among peers, in *PRIME* the content is encoded with Multiple Description Coding (MDC). MDC organizes the streaming content into several sub-streams where each sub-stream can be independently decoded. The delivered quality to each peer is proportional with the number of independent sub-streams that it receives. MDC coding allows each peer to receive a proper number of sub-streams that can be delivered through its access link bandwidth.

Packet scheduling algorithm: A packet scheduling algorithm should achieve the following two goals: (i) fully utilizing the available bandwidth from each parent peer; and (ii) ensuring in-time delivery of requested packets. The pattern of delivery

of individual packets through the overlay mesh (*i.e.*, the path that a packet traverses to reach from source to each peer) depends on the collective behavior of the packet scheduling algorithm at all participating peers as well as the topology of overlay mesh.

In this study, we employ the following packet scheduling algorithm since it outperforms an array of scheduling algorithms that we examined in an earlier study [6]. Each peer keeps track of the available bandwidth (through passive measurement) and available content at each parent (using periodic reports). Given this information, the scheduling algorithm is periodically (once per Δ) invoked to determine the requested packets from each parent in two steps as follows: First, the scheduler identifies new packets with the highest timestamps that have become available among parents during the last period Δ . These new packets are always requested from parents. Second, a random subset of other missing packets is requested from each parent to fully utilize its bandwidth. To achieve load balancing, if a packet is available at more than one parents, it is requested from a parent that has the lowest ratio of requested packets.

C. Dynamics of Content Delivery

In this subsection, we briefly describe the dynamics of content delivery in PRIME. This in turn provides the required background to understand the behavior of the system in our desired scenarios (*e.g.*, resource-poor environment).

To clearly explain the dynamics of content delivery in PRIME, we first present an organized view of a randomly connected mesh. Peers in a randomly connected and directed mesh can be organized into levels based on their shortest distance from source through the directed overlay as shown in 1. Peers that are one-hop away from source (source's children) are grouped in level 1, peers that are two hops away from source are grouped in level 2, and so on. The number of levels of the overlay (*i.e.*, overlay depth d) depends on the number of peers and the number of children that each peer accommodates (*i.e.*, outgoing degree).

Using the scheduling algorithm that we described earlier, each segment of the content is delivered to individual participating peers in two phases: *diffusion phase* and *swarming phase*. During the diffusion phase, each peer receives any piece of a new segment from its parent in the higher level (closer to the source). Therefore, pieces of a newly generated segment are progressively pulled by peers at different levels. For example, pieces of a newly generated segment are pulled by peers in level 1 after one period (Δ), and then pulled by peers to level 2 after $2*\Delta$ and so on. After d periods, all peers in the overlay have one piece of the new segment. Ideally, each piece of a segment is delivered only once by source. Therefore, the group of peers that receive a piece of the segment during the diffusion phase form a subtree that is rooted in a peer in level 1 and is called the diffusion subtree. Shaded nodes in Fig. 1 form a diffusion subtree. All connections from peers in level i to their child peers in level $i + 1$ are called *diffusion connections*. These connections are located on a diffusion subtree.

During the swarming phase, each peer receives all the missing pieces of a segment from its parent in the same or lower levels (further from source). These parents are called swarming parents. The swarming phase may take more than one period since swarming parents may not have all the missing pieces of the segment. Except for the diffusion connections, all other connections in the overlay mesh are swarming connections. The collection of swarming connection forms a directed mesh that we call the swarming mesh. The swarming mesh is used to exchange different pieces of each segment between different diffusion subtrees.

In summary, each piece of any new segment is diffused through a particular diffusion subtree during the diffusion phase of that segment. Then, the available pieces are exchanged between peers in different diffusion subtrees through the swarming mesh during the swarming phase of the segment.

There are two possible bottleneck in content delivery that could limit the delivered quality to individual peers in PRIME [6]: (i) bandwidth bottleneck, and (ii) content bottleneck. The bandwidth bottleneck occurs when the aggregate available bandwidth to a peer is less than its incoming access link bandwidth. The content bottleneck occurs when parent peers do not have sufficient useful content to fully utilize their available bandwidth.

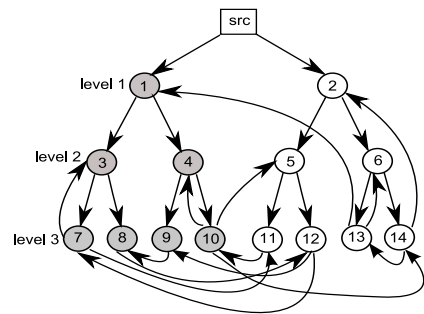


Fig. 1. The organized view of a randomly connected and directed mesh reveals the diffusion subtrees and the swarming mesh

III. EVALUATION

In this section, we examine the effect of available resources, heterogeneity of access link bandwidth and free-loaders on the performance of PRIME using ns simulations.

A. Simulation Settings

In our simulations, we use 200 peers, and the network topology is generated using *Brite* [7] with 15 AS and 10 routers every AS in top-down mode. *RED* queue management is employed at all routers. The delay on the access links are randomly selected between 5ms to 25ms, and the delay of the core links are set by *Brite* (in the range of milliseconds). All pairwise connections between peers employ *RAP* congestion control mechanism [8]. In our simulations, core links have sufficient bandwidth, thus congestion only occurs at the edge of the network. The video stream is MDC encoded with 5

Uplink BW	SC1	SC2	SC3	SC4	S5	SC6
128kbps	27%	54%	13%	5%	11%	50%
384kbps	60%	20%	80%	9%	14%	39%
1000kbps	13%	26%	7%	36%	25%	11%
0kbps	0%	0%	0%	50%	50%	0%
RI	1	1	1	1	0.8	0.8

TABLE I

SIX TARGET SCENARIOS WITH PEERS OF DIFFERENT OUTGOING ACCESS LINK BANDWIDTH

descriptions. For simplicity, we assume that all descriptions have the same constant bit rate of 80 Kbps. Therefore, the rate of full quality version of the stream is 400 Kbps. The incoming access link bandwidth of all peers are set to 550 Kbps so that each peer can easily receive the full quality playback rate.

To examine the effect of aggregate available resource, bandwidth heterogeneity and free-loaders we control the outgoing access link bandwidth of participating peers as follows: The outgoing bandwidth of individual peers can be set to one of the four values: 128 Kbps, 384 Kbps, 1000 Kbps and 0 Kbps. By controlling the distribution of peers across these four groups, we can control the heterogeneity of outgoing access link bandwidth, percentage of free-loaders (with outgoing bandwidth of zero) which in turn determines the aggregate outgoing bandwidth (*i.e.*, system capacity) for a given scenario. Each column in Table I shows the distribution of 200 peers across different groups which represents one of our target scenarios.

We define the *resource index (RI)* as the ratio of the aggregate uplink bandwidths contributed by all peers and source to the amount of required bandwidth for delivery of full quality stream to all peers (*i.e.*, 200*400Kbps). RI quantifies the relative availability of aggregate resources in a given scenario. For example the RI in a scenario where all peers contribute as much bandwidth as they consume, would be one. The last row of Table I presents the value of RI for each target scenario.

We decouple bandwidth and content bottlenecks in our simulations as follows. Each parent peer sends fixed size packets to each children through a congestion controlled connection. If there is no outstanding packet to send (*i.e.*, content bottleneck occurs), the parent sends the same size packet with a special flag. This enables each child peer to determine what portion of the available bandwidth from each parent was not utilized for delivery of data packets. The ratio of all data packets to all delivered packets from all parents represents the *utilization of available bandwidth* and quantifies the degree of content bottleneck observed by each child peer. Furthermore, the aggregate utilization of access link bandwidth also represents the degree of bandwidth bottleneck that is experienced by each child peer.

B. Effect of Available Resource

To examine the effect of available resource, we consider a scenario where all participating peers have the same outgoing bandwidth and then change the value of outgoing access link bandwidth to vary the value of RI from 0.6 to 2.

Fig. 2(a) depicts the *averaged* delivered quality (in terms of number of description) to participating peers as a function of RI. This figure clearly shows that when $RI < 1$, the average delivered quality is less than maximum quality but it increases with RI. Peers receives the full quality in average as long as RI is larger then one. When the resource index is less than one, the received quality is proportional to the resource index.

Fig. 2(b) provides a more detailed view by showing the distribution of delivered quality to individual peers for different RI values. When RI is greater than one, a majority of peers receive full quality stream. But when RI drops below one, the distribution of delivered quality becomes more skewed among participating peers. For instance, when RI is 0.6, the range of delivered quality is between 2.3 to 4 descriptions while all peers have the same incoming bandwidth.

To identify the underlying causes for the skewness of delivered quality when RI is less than 1, we examine the distribution of access link bandwidth utilization among peers as shown in Fig. 2(c). This figure shows that as RI drops below 1, the utilization of access link bandwidth becomes skewed similar to delivered quality. This in turn indicates that the bandwidth bottleneck is the main factor in the diversity of delivered quality to different peers. More specifically, when $RI < 1$, some peers luckily obtain a larger portion of available resources because their aggregate bandwidth from their parents is higher due to the dynamics of congestion control mechanism.

C. Effect of Heterogeneous Bandwidth

We now examine the impact of heterogeneous uplink bandwidth on the performance of P2P streaming. Towards this end, we focus on scenarios SC1, SC2 and SC3 in Table I. While these three scenarios have the same RI value of 1, their distribution of uplink bandwidth is different. Fig. 3(a) and Fig. 3(b) depict the CDF of delivered quality and utilization of access link bandwidth among participating peers for these three scenarios. These figures show that the degree of heterogeneity in uplink bandwidth does not affect the distribution of delivered quality to participating peers.

We also examine the correlation between the delivered quality and the contributed resources (*i.e.*, outgoing bandwidth) of participating peers. Fig. 3(c) depicts a scattered plot of the delivered quality to individual peers as a function of their uplink bandwidth. This figures illustrates that there is no correlation between the contributed resource by each peer and the quality it receives (*i.e.*, the amount of resources it consumes). This is clearly an improper behavior since it does not provides an incentive for higher bandwidth peers to participate. To address this issue, P2P streaming mechanism should incorporate a contribution-aware resource allocation mechanism such that the allocated resource to individual peer would be proportional with their contributed resources.

D. Effect of Freeloaders

To investigate the impact of free loaders, we examine scenarios SC4 and SC5 with RI values of 0.8 and 1, respectively, and half of participating peers are free-loaders as shown in

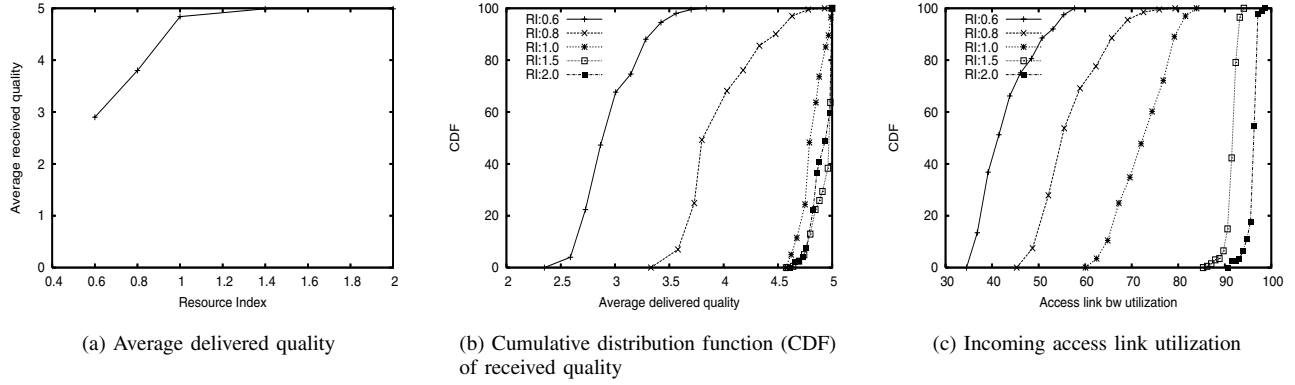


Fig. 2. Effect of available resources with homogeneous uplink bandwidth

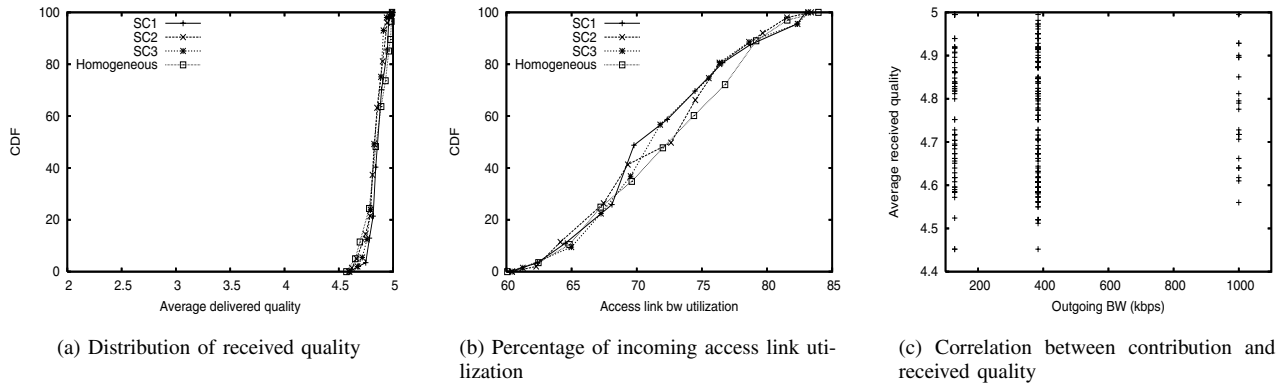


Fig. 3. Effect of heterogeneous uplink bandwidth

Table I. Fig. 4 shows the distribution of delivered quality to participating peers in scenarios SC4 and SC5. We have also included the distribution of delivered quality for peers with homogeneous outgoing bandwidth when RI is 0.8 and 1 as references for comparison. Fig. 4 reveals that the presence of free-loaders significantly reduces the delivered quality even when the aggregate available resources (*i.e.*, the value of RI) remains intact. We note that the scenario with free-loaders can be viewed as a special case for bandwidth heterogeneity. Therefore, the significant drop in delivered quality as the result of free-loader was rather surprising since the heterogeneity of bandwidth does not have a major effect on performance as we reported in the previous subsection.

A closer examination of our results revealed that the free-loaders affect the connectivity of the overlay in such a way that they disrupt the two phase content delivery. This behavior can be explained as follows: Since free-loaders have no child peers, their presence in the overlay can affect the connectivity (and thus exchange of content) between different diffusion subtrees. For example, if peer 7, 8, 9, and 10 in Fig. 1 are free-loaders their corresponding diffusion subtree will not be connected to the swarming mesh (does not have any swarming connection).

More specifically, a diffusion subtree that only has free-loaders at its bottom level cannot provide its corresponding content to peers on other diffusion subtrees (*e.g.*, peers 5, 11, 12 and 14 in Fig. 1). This in turn limits the delivered quality to other participating peers. However, peers on such a disconnected diffusion subtree can still receive the content from other diffusion subtrees due to the directed nature of connectivity in the overlay. Even if the diffusion subtrees do not get completely disconnected from the swarming mesh, the presence of free-loaders could increase the depth of the overlay which in turn affects the buffer requirement at each peer [6]. *In summary, our results reveal that the presence of free-loaders can significantly affect the connectivity between different diffusion subtrees in the overlay which in turn prevents content swarming among them and thus limits the delivered quality to a subset of peers.* This suggests that P2P streaming mechanism should ensure proper connectivity among participating peers (*e.g.*, monitoring the overlay) to address such significant drop in quality. Clearly, identifying free-loaders and removing them from the overlay would also help but it is non-trivial in such a distributed environment without central authority.

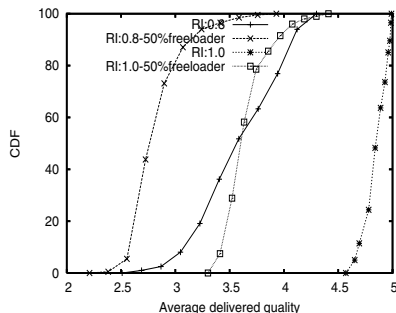


Fig. 4. Effect of free-loaders on delivered quality

E. Effect of the Number of Users

Finally we examine the impact of group size on system performance. Fig. 5 depicts the average delivered quality among participating peers as a function of number of peers in the system in two scenarios SC1 and SC6 with heterogeneous uplink bandwidth and resource index 1 and 0.8, respectively. This figure illustrates that the group size does not have any effect on the performance of *PRIME* regardless of the distribution of uplink bandwidth. The primary factor that determines the delivered quality is the value of RI.

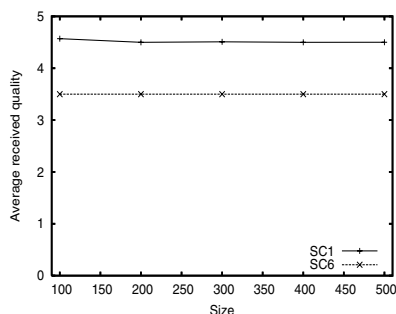


Fig. 5. Effect of number of users on average delivered quality

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we examine the potential issues and challenges in offering P2P streaming service to residential users. In particular, we focus on the impact of resource availability, outgoing bandwidth heterogeneity and free-loaders on the performance of P2P streaming. We identified two potential problems. First, in resource-poor scenarios, the delivered quality to individual peers is not correlated with their contributed resource. Second, while P2P streaming mechanisms are able to effectively accommodate the heterogeneity of uplink bandwidth, the presence of free-loaders could significantly affect the connectivity of the overlay and thus the feasibility of data exchange among peers. This in turn could significantly degrade the delivered quality to a subset of peers in the system.

To successfully deploy a P2P streaming service to residential users, our findings suggest that these services should be able to address the following issues: (i) *Coping with free-loaders*: the overlay construction mechanism should be able to

identify and cope with free-loaders in the overlay to minimize their impact on other peers, (ii) *Contribution-aware resource allocation*: the content delivery mechanism should incorporate a scheme that controls the allocated resources (and thus the delivered quality) to individual peers based on their contributed resources to the system.

As part of our future plans, we are exploring various techniques to address the above two issues in offering P2P streaming service to residential users.

REFERENCES

- [1] B. Cohen, "Incentives built robustness in bittorrent," in *1st Workshop on the Economics of peer-2-peer systems*, June 2003.
- [2] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "Coolstreaming: A data-driven overlay network for live media streaming," in *IEEE INFOCOM*, 2005.
- [3] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *IEEE ICNP*, 2003.
- [4] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. E. Mohr, "Chainsaw: Eliminating Trees from Overlay Multicast," in *International Workshop on Peer-to-Peer Systems*, 2005.
- [5] X. Jiang, Y. Dong, D. Xu, and B. Bhargava, "GnuStream: A P2P Media Streaming System Prototype," in *IEEE International Conference in Multimedia and Expo*, 2003.
- [6] N. Magharei and R. Rjaie, "PRIME: Peer-to-Peer Receiver-driven MESH-based Streaming," tech. rep., 2006.
- [7] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation," in *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, 2001.
- [8] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *IEEE INFOCOM*, 1999.