PRIME: Peer-to-Peer Receiver-Driven Mesh-Based Streaming

Nazanin Magharei and Reza Rejaie, *Senior Member, IEEE*

***Abstract*—The success of file swarming mechanisms such as BitTorrent has motivated a new approach for scalable streaming of live content that we call *mesh-based* Peer-to-Peer (P2P) streaming. In this approach, participating end-systems (or peers) form a randomly connected mesh and incorporate swarming content delivery to stream live content. Despite the growing popularity of this approach, neither the fundamental design tradeoffs nor the basic performance bottlenecks in mesh-based P2P streaming are well understood.**

In this paper, we follow a performance-driven approach to design PRIME, a scalable mesh-based P2P streaming mechanism for live content. The main design goal of PRIME is to minimize two performance bottlenecks, namely *bandwidth bottleneck* and *content bottleneck*. We show that the global pattern of delivery for each segment of live content should consist of a *diffusion* phase which is followed by a *swarming* phase. This leads to effective utilization of available resources to accommodate scalability and also minimizes content bottleneck. Using packet level simulations, we carefully examine the impact of overlay connectivity, packet scheduling scheme at individual peers and source behavior on the overall performance of the system. Our results reveal fundamental design tradeoffs of mesh-based P2P streaming for live content.

***Index Terms*—Communication systems, computer networks, multimedia communication, multimedia systems, Internet.**

I. Introduction

PEER-TO-PEER (P2P) overlays offer a promising approach to stream *live* video from a single source to a large number of receivers (or peers) over the Internet without any special support from the network. This approach is often called *P2P streaming*. The goal of P2P streaming mechanisms is to deliver high quality stream to individual peers in a scalable fashion. To gracefully scale with the number of participating peers in a session, a P2P streaming mechanism should be able to effectively utilize the contributed resources (namely outgoing bandwidth) of individual peers. Achieving this goal is challenging due to the heterogeneity and asymmetry of access link bandwidth as well as the dynamics of participation (i.e., churn) among peers.

A well known approach to P2P streaming is organizing participating peers into multiple, diverse tree-shaped overlays where each specific "sub-stream" of the live content is *pushed* through a particular tree from source to all interested peers (e.g., [1]). This approach has two important limitations: (*i*) in the presence of churn, maintaining multiple tree-shaped overlays with desired properties could be very challenging [2]. (*ii*) the rate of content delivery to each peer through individual trees is limited by the minimum throughput among the upstream connections which could be even smaller than the bandwidth of a single sub-stream [2].

Recently, the success of file swarming mechanisms (e.g., BitTorrent) has motivated another approach to P2P streaming that we call *mesh-based P2P streaming*. In this approach participating peers form a mesh-shaped overlay and incorporate *swarming* (or pull) content delivery. File swarming mechanisms (e.g., [3], [4]) leverage the elastic nature of the content and the availability of the entire file at the source to effectively utilize available resources and scale. More specifically, in file swarming mechanisms source distributes different pieces of a file among participating peers which enables them to exchange their pieces and actively contribute their outgoing bandwidth. Individual peers pull different segments of the file in a pseudo-random order and potentially at different rates from their neighbors in the overlay. Incorporating swarming content delivery into mesh-based P2P streaming mechanisms for "live" content is challenging for two reasons: (*i*) Ensuring the in-time delivery for individual packets of streaming content is difficult. (*ii*) Since the content is progressively generated by a live source, the availability of new content for delivery is limited. This reduces the diversity of available pieces among participating peers which in turn degrades the utilization of their outgoing bandwidth.

As we discuss in Section II, a few mesh-based P2P streaming mechanisms have been recently proposed [5]–[10]. However, to our knowledge, none of these studies have answered the following important questions:

- How can swarming content delivery be incorporated into a mesh-based P2P streaming mechanism for live content to effectively scale with peer population?
- What are the fundamental tradeoffs and limitations in design of such a scalable mesh-based P2P streaming mechanism for live content?

The first contribution of this paper is to address these two important questions. Towards this end we design *PRIME*, a new mesh-based P2P streaming mechanism for delivery of live content. We follow a *performance-driven* approach to design PRIME. Initially we identify two performance bottlenecks in mesh-based P2P streaming that could limit the utilization of available resources and thus limit the scalability as follows: (*i*) A peer experiences *bandwidth bottleneck* when its aggregate rate of content delivery from its neighbors is not sufficient to fully utilize its incoming access link bandwidth. (*ii*) A peer experiences *content bottleneck* when there is not

Manuscript received December 25, 2007; revised June 27, 2008; approved by IEEE/ACM Transactions on Networking Editor M. Hofmann. First published March 21, 2009; current version published August 19, 2009. This material is based upon work supported by the National Science Foundation under NSF CAREER Award CNS-0448639. An earlier version of this paper appeared in the Proceedings of IEEE INFOCOM 2006.

The authors are with the Department of Computer and Information Science, University of Oregon, Eugene, OR 97403 USA (e-mail: nazanin@cs.uoregon.edu; magharei@cs.uoregon.edu; reza@cs.uoregon.edu).

Digital Object Identifier 10.1109/TNET.2008.2007434

sufficient amount of useful content among its neighbors to effectively utilize its available bandwidth from them. We show that the probability of bandwidth bottleneck directly depends on the connectivity of the overlay (i.e., the incoming and outgoing degrees of individual peers). We then derive the proper connectivity for individual peers that minimizes the probability of bandwidth bottleneck among them.

We show that the probability of content bottleneck among peers directly depends on the global pattern of content delivery from source to all peers in the overlay. We introduce the "organized view" of a random mesh and then derive the desired pattern of content delivery for a single segment that minimizes the probability of content bottleneck among peers and thus maximizes the utilization of resources to accommodate scalability. We demonstrate that the desired pattern of delivery should consist of two phases: (*i*) a *diffusion* phase where data rapidly flows away from source, and is followed by (*ii*) a *swarming* phase where peers exchange their available packets. We derive the required "packet-pulling" strategy at individual peers that its collective behavior across all peers leads to the desired pattern of delivery. The two-phase view of the content delivery leads to two important insights: (*i*) It reveals the impact of overlay connectivity and source behavior on the performance of content delivery. (*ii*) It demonstrates some fundamental limitations of the system by illustrating the relation between peer population, overlay connectivity and minimum buffer requirement at individual peers.

The second contribution of this paper is the detailed performance evaluations of PRIME using packet level simulations. We show that the notion of diffusion and swarming phases offers a powerful method to identify the performance bottlenecks of a mesh-based P2P streaming mechanism. We carefully examine the performance of PRIME in scenarios with limited resources and untangle the effect of different parameters on overall performance of PRIME. Our results not only reveal a few fundamental design tradeoffs and limitations in incorporating swarming content delivery into mesh-based P2P streaming for live content but also shed an insightful light on the dynamics of swarming content delivery in these systems. Some of our main findings can be summarized as follows:

(i) Ensuring the same ratio of bandwidth to degree among participating peers minimizes the bandwidth bottleneck in the overlay.

(ii) There is a sweet range for peer degree over which swarming content delivery exhibits a good performance and effectively scales with peer population. The lower bound of this range is 6 but the upper bound is determined by peer bandwidth.

(iii) The minimum buffer requirement at each peer is directly proportional to the total duration of the diffusion and swarming phases for each packet. The minimum duration of diffusion phase depends on the depth of the overlay whereas the minimum duration of swarming phase depends on the connectivity of the overlay. Bi-directional overlays require larger buffering at individual peers due to the lower diversity in connectivity which adversely affects swarming content delivery.

(iv) In a properly connected overlay with the sufficient amount of resources (i.e., aggregate outgoing bandwidth among peers is not smaller than their aggregate incoming bandwidth) neither the heterogeneity and asymmetry of access link bandwidth nor the location of high bandwidth peers significantly affects the delivered quality to individual peers. However, the presence of free-riders may limit the connectivity between regions of the overlay and thus prevent the delivery of a subset of packets to some regions of the overlay.

(v) The packet scheduling scheme at individual peers should pull any newly generated packets (with the highest timestamps) from parents to ensure proper diffusion of content through the overlay. Besides this requirement, the actual criteria for selecting packets from individual parents does not have a significant impact on the performance of content delivery as long as load is properly balanced among parents.

(vi) Incorporating some light weight coordination mechanism (i.e., careful packet swapping and loss detection) at source can significantly improve overall performance of content delivery.

(vii) The more imbalanced the bandwidth-degree ratio among participating peers (i.e., the more distorted the overlay) becomes, the lower the diffusion rate of new packets through the overlay becomes, and the lower the delivered quality to individual peers would be.

The rest of this paper is organized as follows. We briefly describe related studies in Section II. In Sections III and IV, we describe two key components of PRIME, namely overlay construction and content delivery mechanisms, respectively. Section V presents simulation-based evaluations of PRIME and illustrates some of the key tradeoffs and limitations in the design of mesh-based P2P streaming for live content. Section VI concludes the paper and sketches our future plans.

II. Related Work

In this section, we focus on a few previous studies that are most related to our work. CoopNet [1] and SplitStream [11] both organize peers into multiple, diverse trees and push each sub-stream of the content through a specific tree. This enables all peers to contribute their outgoing bandwidth and also limits the impact of a peer departure to a single tree. In our recent study [2], we compare multi-tree and mesh-based P2P streaming approaches and show that (*i*) in the presence of churn, maintaining multiple trees with desired properties is challenging, and (*ii*) the delivered quality in multi-tree approach is very sensitive to variation in throughput of individual connections.

ChunkySpread [12] is a more recent multi-tree approach to P2P streaming. ChunkySpread uses frequent signaling among peers to achieve load balancing and latency reduction by adaptively changing their parents while avoiding loops within each tree. Authors focus on the design and evaluations of multiple trees in resourceful environments. However, the performance of actual content delivery in the presence of packet level dynamics (and loss) and the impact of overlay properties (e.g., node degree, peer bandwidth) have not been explored.

CoolStreaming/DONet [13] is a mesh-based approach where peers initially form a mesh [14]. However, once each peer identifies proper parents, it requests each parent to provide a specific sub-stream of the content. In essence, CoolStreaming eventually organizes peers into multiple trees and incorporates push-based content delivery [15]. Using prototype implementation, authors conduct experiment over PlanetLab and report on their experience with large scale deployment of this system. Authors present average delivered quality to the peers as a function of peer degrees (over a small range from 2 to 6) and churn. While this study clearly demonstrates the scalability of mesh-based P2P streaming, it does not demonstrate the fundamental tradeoffs in the design of mesh-based P2P streaming mechanisms.

Several studies have proposed to add the notion of "delivery window" to Bittorrent in order to support "streaming" content delivery (e.g., [5], [6], [16]). These studies appear to be targeting playback streaming or on-demand applications. One important difference between live and on-demand P2P streaming is the availability of content for swarming content delivery. In VoD applications the entire content is usually available which increases the diversity of available content among peers and accommodates swarming content delivery. However, in the context of live P2P streaming applications such as PRIME, accommodating swarming content delivery is more challenging because the useful content for swarming is being gradually generated by the source and is more limited. Therefore, the performance of the proposed on-demand P2P streaming mechanisms with limited available content and limited resources is unknown. Finally, a growing number of P2P streaming systems (e.g., wwitv.com, sopcast.com) have become available for broadcasting the streaming content to a large group of end-systems over the Internet. However, no technical details about these systems is available for comparison.

In this paper, we primarily focus on the effect of swarming content delivery (i.e., packet scheduling) and overlay connectivity on the performance of mesh-based P2P streaming mechanisms for live content. We also explore the underlying causes of observed behavior, and identify fundamental tradeoffs and limitations of mesh-based P2P streaming. To our knowledge, none of the previous studies have achieved these goals.

III. Overlay Construction in PRIME

Participating peers in PRIME maintain a *randomly* connected and *directed* overlay (i.e., a mesh-shaped overlay). Such an overlay is easy to maintain and very resilient to churn. Furthermore, incoming and outgoing connections of each peer are more likely to have diverse paths which in turn reduces the probability of a shared bottleneck among them. There is a parent-child relationship between connected peers and content is always delivered from the parent the child. Each peer maintains connections from multiple parents and serves multiple children. All connections are initiated by children. When a peer needs more parent(s), it contacts a bootstrapping node to learn about a random subset of other participating peers in the system and then requests some of those peers to serve as its parent. We note that PRIME can certainly incorporate other (distributed or central) peer discovery and parent selection techniques. However, as long as the incoming and outgoing degrees of individual peers are not affected, other details of these techniques do not have any significant impact on PRIME performance.

To construct the overlay, each peer tries to maintain a sufficient number of parents that can collectively fill its incoming access link bandwidth. All connections in the overlay are congestion controlled (using RAP [17] or TFRC [18]). The key design question for the overlay construction mechanism is "*how to determine the incoming and outgoing degree of individual peers?*"

Deriving Proper Peer Degree: Suppose that each peer always has sufficient amount of useful content to send to its children. Then, the aggregate rate of content delivery to each peer depends not only on its number of parents (i.e., incoming degree) but also on the number of children (i.e., outgoing degree) for each one of its parents. Without loss of generality, we assume that congestion only occurs at the edge of the network, i.e., at the incoming or outgoing access links of participating peers. Therefore, the average bandwidth for a congestion controlled connection between parent p to child c can be roughly estimated as $MIN(outbw_p/outdeg_p, inbw_c/indeg_c)$ where $outbw_p$, $outdeg_p$, $inbw_c$, $indeg_c$ denote the outgoing bandwidth and outgoing degree of peer p, and incoming bandwidth and incoming degree of peer c, respectively. If $(outbw_p/outdeg_p) < (inbw_c/indeg_c)$, the outgoing access link of the parent is the bottleneck and thus the incoming access link of the child may not be fully utilized. In contrast, if $(outbw_p/outdeg_p) > (inbw_c/indeg_c)$, the bottleneck is at the incoming access link of the child and the outgoing access link of the parent may not be fully utilized.

This observation suggests that to avoid a significant bottleneck at both incoming and outgoing access link of all peers in a randomly connected overlay, the same ratio of "bandwidth to degree" should be used for the outgoing and incoming connections of *all* peers. More specifically, any two randomly selected peers i and j in the overlay should satisfy the following condition:$bwpf = outbw_i/outdeg_i = inbw_j/indeg_j$.

We call this *bandwidth-degree condition*. This condition implies that all connections in the overlay have roughly the same bandwidth of $bwpf$, or bandwidth-per-flow. In essence, $bwpf$ is a configuration parameter that directly translates the (potentially heterogeneous and asymmetric) access link bandwidth of individual peers (and the source) to their proper incoming and outgoing degree.

To illustrate the effect of bandwidth-degree condition on the utilization of access link bandwidth, we conduct ns simulations where 200 peers with symmetrical access link bandwidth of bw_h or bw_l form a directed and randomly connected mesh. All peers use the same incoming and outgoing degree regardless of their bandwidth. Connections are congestion controlled using RAP [17]. Fig. 1 depicts the average utilization of incoming access link bandwidth and its 10th and 90th percentiles (as bar) only among high bandwidth peers (bw_h) for two levels of bandwidth heterogeneity where bw_h/bw_l is equal to 2 and 8. We examine each level of bandwidth heterogeneity with three different values of peer degree (namely 8, 12 and 16), and different fraction of high bandwidth peers (n_h) for each degree. Across all these 18 scenarios, the incoming access link of low bandwidth peers has always been utilized. Fig. 1 indicates that if all peers use the same degree, increasing the degree of bandwidth heterogeneity decreases the average utilization of access link bandwidth among high bandwidth peers especially when the fraction of high bandwidth peers is small (e.g., $bw_h/bw_l = 8$ and $n_h = 10\%$). Setting the peer degree based on the bandwidth-degree condition in all these scenarios results in a high utilization (>95%) of access link bandwidth among all peers with low variations (<3%) in all the above scenarios. The uti-

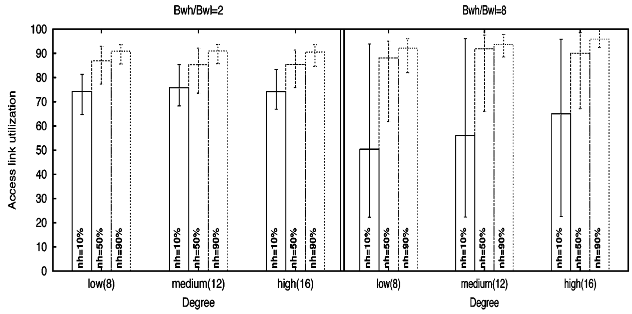


Fig. 1. Utilization of access link bandwidth across different peer degree and various level of heterogeneity, when all peers have the same incoming and outgoing degree regardless of their bandwidth.

lization of access link bandwidth in those settings where the bandwidth-degree condition is satisfied, is not shown in Fig. 1 for clarity. In summary, accommodating the bandwidth-degree condition ensures that each peer can receive content at the maximum rate and does not experience a *bandwidth bottleneck*.

In practice, the observed bandwidth for congestion controlled connections in the overlay is likely to be different due to the difference in their round-trip-time or loss rate. Furthermore, some connections might experience bottleneck in the core rather than the edge of the network. This may affect the utilization of access link bandwidth for the children that receive content through these connections. This problem can be addressed by incorporating an adaptation scheme that (*i*) allows children with low utilization of incoming access link bandwidth to have extra parents and (*ii*) allows parents with poor utilization of outgoing access link bandwidth to accept extra children beyond the limit that is specified by the bandwidth-degree condition. We note that the above adaptation scheme should be used for minor tuning of incoming/outgoing peer degree and can not replace the bandwidth-degree condition. Given the dependency of congestion control bandwidth of individual connections to the degree of corresponding peers, the degree of each peer affects not only its own bandwidth utilization but also the bandwidth utilization of its children or parent peers. If peers independently try to determine their proper incoming/outgoing degree, the ripple effect of this decision could easily lead to instability of the overlay. The bandwidth-degree condition provides an implicit coordination for individual peers to determine their degree in a coherent fashion and thus avoids any oscillations in the overlay.

IV. Content Delivery in PRIME

PRIME incorporates swarming content delivery which combines *push* content reporting by parents with *pull* content requesting by children. Each peer simultaneously receives content from *all* of its parents and provides content to *all* of its children. Each peer, as a parent, progressively reports the availability of its new packets to all of its children. Given the available packets at individual parents, a *packet scheduling* scheme at each peer periodically (i.e., once per Δ second) determines an ordered list of packets that should be requested from each parent Parents simply deliver requested packets by each child in the provided order and at the rate that is determined by the congestion control mechanism. We assume that the content is encoded with Multiple Description Coding (MDC). While this is not a requirement for PRIME, it enables each peer to receive a quality proportional to its incoming bandwidth by pulling a proper number of descriptions.

In the context of live P2P streaming applications, source progressively generates a new segment of content once every Δ seconds where a segment consists of a group of packets with consecutive timestamps ($[t_{src} - \Delta, t_{src}]$) across all descriptions, and t_{src} denotes source's playout time. To effectively accommodate swarming, peers should maintain a loosely synchronized playout time which is $\omega * \Delta$ seconds behind source's playout time. Maintaining synchronized playout time maximizes the overlap among buffered data at different peers by providing roughly $\omega * \Delta$ seconds worth of content that can be swarmed among peers. This also facilitates parent selection because each peer with open slot can serve as a parent.[1] The relative playout delay between the source and peers has two implications: (*i*) each peer should buffer at least $\omega * \Delta$ seconds worth of content, and (*ii*) each packet should be delivered within $\omega * \Delta$ seconds from its generation time to ensure in-time delivery.

Avoiding Content Bottleneck: Suppose all connections have roughly the same bandwidth ($bwpf$), then the maximum amount of data that a child can receive from a parent during an interval (Δ) is equal to $D = bwpf * \Delta$. This amount of data is called a *data unit* and consist of several packets (possibly from different descriptions) that are selected by the packet scheduling scheme at a child. When one (or multiple) parent(s) of a child do not have a data unit worth of new content to deliver during an interval, the child cannot fully utilize the bandwidth from the corresponding connection(s) and experiences *content bottleneck*.

The goal of the packet scheduling scheme at individual peers is to maximize their delivered quality with minimum buffer requirement. This goal can be achieved by minimizing the probability of content bottleneck among peers which in turn maximizes the utilization of the outgoing bandwidth among all peers and thus improves scalability. The probability of content bottleneck among peers (i.e., the availability of new data units at individual parents) directly depends on the global pattern of content delivery from the source to all peers through the overlay. Therefore, to design a scalable P2P streaming mechanism, first we identify the global pattern of content delivery that minimizes the probability of content bottleneck among peers. Then, we derive the required packet scheduling scheme at individual peers that leads to the desired global pattern.

A. Organized View of a Random Mesh

To identify the desired global pattern of content delivery, first we present an organized view of a randomly connected and directed mesh. Towards this end, we define the distance of peer p from the source as the length of the shortest path (in hops) from the source to peer p through the overlay. Then, peers that have the same distance of n hops from source can be grouped into *level* n, as shown in Fig. 2.

[1]While this may seem intuitive, some of the P2P streaming mechanisms [9] have assumed that a peer has to delay its playout compare to its parents to provide more time for content delivery. This approach could lead to a long delay between source and some peers, and would limit the choices of parents to only those peers that have earlier playout time.

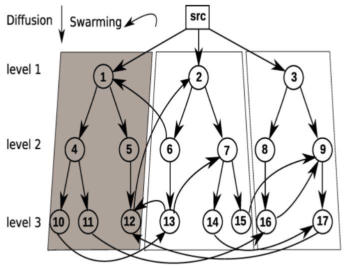


Fig. 2. Organized view of a mesh-based overlay with 17 peers, forming three diffusion subtrees. For clarity, only a subset of connections are shown.

Consider an overlay with P homogeneous peers where all peers have the same incoming and outgoing degree of deg and the source degree of deg_{src}. The organized view reveals three important properties of this overlay as follows [19]: (*i*) the population of peers at level n (or $pop(n)$) is limited to $pop(n) \leq deg_{src} * deg^{(n-1)}$, (*ii*) a lower bound for the number of levels, or $depth$, of such an overlay is $depth \geq log_{deg}(P/deg_{src})$, (*iii*) for a randomly selected peer in the overlay, the probability of having a parent at level n is equal to $pop(n)/P$. Typically, a peer in level n, except for peers in the bottom level, has a single parent in level $n-1$, $(deg-1)$ parents in the same or lower levels, and deg children in level $n+1$. Peers in the bottom level $(n = depth)$ often have a single parent in level $n-1$, and deg children in the same or higher levels.

B. Pattern of Delivery for a Single Segment

In this subsection, we derive the global pattern of content delivery for a single segment of content that minimizes the probability of content bottleneck among peers. Consecutive segments of the stream can be delivered through the overlay using a roughly similar pattern. Intuitively, to minimize the number of intervals for delivery of a segment, first different data units of the segment should be rapidly delivered (or diffused) to different subset of peers. Then, peers can exchange (or swarm) their data units and contribute their outgoing bandwidth until each peer has a proper number of data units for that segment. This observation motivates a two-phase approach for the delivery of a segment as follows:

1) Diffusion Phase of Delivery: Once a new segment becomes available at the source, peers in level 1 can collectively pull all data units of that segment during the next interval Δ. Then, peers in level 2 can collectively pull all data units of the new segment during the following interval and so on. Therefore, it takes $depth * \Delta$ seconds until at least one data unit of a newly generated segment (by source) reaches (i.e., diffuses to) each peer in the system. We call this *diffusion time* of a segment.

To rapidly diffuse a new segment to peers in lower levels of the overlay, all the connections between peers in level n $(n < depth)$ to their children in level $n+1$ should be exclusively used for the diffusion of new data units. These connections are called *diffusion connections* and the corresponding parents are called *diffusion parents*. Diffusion connections are shown with straight arrows in Fig. 2. The number of diffusion connections into level n is at least equal to the population of peers in level n (i.e., $deg_{src} * deg^{(n-1)}$) which is exponentially increasing with n.

The above pattern of content delivery has the following implications: First, the diffusion phase of a segment takes exactly $depth$ intervals or $depth * \Delta$ seconds. Second, each peer p in level 1 as well as all of its descendant peers in a sub-tree rooted in p receive the same data unit of each segment during the diffusion phase of that segment, but at different intervals depending on their levels. Each such a sub-tree of peers that is rooted in a peer in level 1 is called a *diffusion sub-tree*. The number of diffusion sub-trees in an overlay is equal to the population of peers in level 1, or deg_{src}. In Fig. 2, one of the three diffusion sub-trees that is rooted at peer 1, is shaded. Third, when the bandwidth of a diffusion connection is less than $bwpf$, all the downstream peers in the corresponding diffusion sub-tree experience content bottleneck during the diffusion phase. *We emphasize that the diffusion sub-trees are implicitly formed as a result of pull packet scheduling by individual peers.*

2) Swarming Phase of Delivery: At the end of the diffusion phase of a segment, all peers in the overlay have at least one data unit of that segment. During the swarming phase of a segment, peers pull the missing data units of the segment from their parents that are located in the same or lower levels. Therefore, all the connections from parents in level j to their children in the same or higher level $i (i \leq j)$ should be exclusively utilized for swarming. These connections are called *swarming connections* (shown with curly arrows in Fig. 2) and their corresponding parents are called *swarming parents*. Given the distribution of peers at different levels of the overlay, almost all the swarming parents are located at the bottom level. This means that the outgoing bandwidth of peers at the bottom level is primarily utilized for the swarming of individual segments.

We recall that all peers in the same diffusion sub-tree receive the same data unit of a segment during the diffusion phase. This implies that only those swarming parents that are located on different diffusion sub-trees can immediately provide a new data unit to a child at the end of the diffusion phase. For example, in Fig. 2, p_9 can immediately obtain a new data unit from p_{15} but not from p_{16}. If all the swarming parents of a child i are located on different diffusion sub-trees, the child can pull $(indeg_i - 1)$ new data units from all parents in a single swarming interval (e.g., p_{12} in Fig. 2). Otherwise, the child experiences a content bottleneck (e.g., p_9 in Fig. 2) and thus requires more than one swarming interval to obtain the remaining data units. During these extra intervals, some of its swarming parents will obtain new data units of the target segment, and can pass them along in the following interval. For example, p_{16} receives a new data unit from p_{11} after one interval and can pass it to p_9 in the next interval.

In a randomly connected overlay, the probability of experiencing a content bottleneck during the swarming phase depends on the relative value of peer's incoming degree and the number of diffusion sub-trees with a unique data unit as well as the population of peers in the bottom level of each diffusion sub-trees. For a given overlay, the minimum number of swarming intervals (or K_{min}) is determined such that a majority of peers can receive their required number of data units (i.e., proper number descriptions) of a segment. In Section V, we show how the value of K_{min} is affected by other system parameters. In summary, the required buffer at individual peers or their relative playout delay compare to source (i.e., $\omega * \Delta$ seconds) should be suffi-

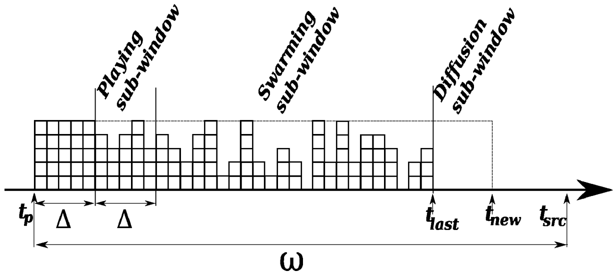


Fig. 3. Buffer state at an scheduling event.

ciently long to accommodate both diffusion and swarming intervals for almost all peers by satisfying the following condition: $(depth + K_{min}) \leq \omega$.

C. Receiver-Driven Packet Scheduling

The packet scheduling scheme at each peer determines requested (i.e., pulled) packets from individual parents. We assume that each packet can be uniquely identified by its description id and a timestamp. The packet scheduling at each peer takes the following input parameters: (*i*) the peer's target quality (i.e., number of descriptions) that are being played (n), (*ii*) the exponentially weighted moving average of congestion controlled bandwidth from each parent $(ewma\_bw(i))$, (*iii*) reported packets by individual parents that are required, and (*iv*) peer's own playout time (t_p) as well as the packets that it has already received (i.e., its buffer state). Given the above information, the packet scheduling scheme at each peer should determine requested packets from each parent in order to maximize the utilization of their available bandwidth. To relate the packet scheduling at each peer with the global pattern of content delivery, we divide the relevant packets at each scheduling event into the following sub-windows based on their timestamps as shown in Fig. 3:

- *Playing Sub-window*: Packets in this sub-window are most likely received and any missing packet should be requested and delivered during the current scheduling event.[2]
- *Swarming Sub-window*: Packets in this sub-window are partially delivered and a random subset of missing packets in this sub-window should also be requested during this scheduling event.
- *Diffusion Sub-window*: This sub-window represents those packets with the highest timestamps that have become available since the last scheduling event. These packets are available only at the diffusion parent(s) and none of these packets have been requested (and thus is not available) yet.

The packet scheduling scheme at each peer is invoked once every Δ seconds and takes the following steps:

I) Quality Adaptation: it compares the average value of aggregate rate of data delivery $(\sum ewma\_dr(i))$ from all parents with the target quality (i.e., the number of requested descriptions). If the aggregate rate of delivery is sufficient to accommodate another description, the target quality is increased by one description, i.e., $IF\, C*(n+1) \leq \sum ewma\_dr(i)\, THEN\, n = n+1$. When the aggregate rate of delivery is not sufficient to sustain the current number of descriptions and the available buffer can not compensate this bandwidth deficit during one interval Δ, the target quality is reduced by one.

II) Requesting Diffusion Packets: the scheduler requests any available packets within the diffusion sub-window until all such packets are requested or the bandwidth of the parent(s) are fully utilized. Note that only diffusion parents have packets within diffusion sub-window. This strategy ensures rapid diffusion of new packets to lower levels of the overlay.

III) Requesting Playing Packets: Any missing packets within the playing sub-window is requested from the parents according to the scheduling and parent selection algorithm described below.

IV) Requesting Swarming Packets: the scheduler requests a subset of packets in the swarming sub-window that are available among parents and needed by the child. The requested packets are determined in two steps as follows: *(i) Selecting Timestamps*: the scheduler determines the number of missing packets for each timestamp within the swarming sub-window by simply comparing the target quality with the number of unique packets (from different descriptions) that it has already received for each timestamp. This step generates a list of timestamps for packets that can be pulled from swarming parents. *(ii) Assigning Packets*: To select a random subset of required packets, the scheduler shuffles the list of selected timestamps and sequentially examines each timestamp by taking two related actions:

- *Description Selection*: Determining a proper description such that the corresponding packet (timestamp, description) is available among parents but missing at the child, and
- *Parent Selection*: Assigning the identified packet to a parent that can provide it and has unused bandwidth.

The description for a given timestamp could be determined by selecting a *random* or *rarest* description from the useful descriptions among parents. The parent can be selected either randomly or based on the minimum ratio of its assigned packets to its total packet budget (i.e., the fraction of its packet budget that has been already assigned). Given the average bandwidth from each parent, we can estimate the total budget of each parent during one interval $((ewma\_bw(i) * \Delta)/PktSize)$. The latter parent selection criteria tends to proportionally balance the assigned packets among parents during the scheduling process. The criteria and ordering for selection of description and parent of each required timestamp result in six variants of the scheduling scheme. We examine these six variants of scheduling in Section V-C.

D. Source Behavior

The maximum available quality in the system is determined by the aggregate quality (i.e., number of descriptions for each timestamp) that are delivered from the source to all of its children in level 1. This quality in turn depends on two factors: (*i*) the aggregate bandwidth from the source to all of its children, and (*ii*) the rate of delivery for new packets from source to peers in level 1 which we call *diffusion rate*. For example, if the same packet is requested (and thus sent) to multiple peers in level 1, the diffusion rate might be significantly lower than the aggregate bandwidth from source. In contrast, if

[2]Packets from t_p till the start of playing sub-window are being played during this interval and should be already available in the buffer.

all packets are unique, the diffusion rate is equal to the aggregate bandwidth from source. We recall that source's outgoing degree is determined by the bandwidth-degree condition to ensure high utilization of its access link bandwidth. Therefore, if source's access link bandwidth is equal to (or larger than) the stream bandwidth, it can deliver the full quality stream to the system if its aggregate bandwidth is properly used. If the diffusion rate is equal to the stream bandwidth, then we observe proper behavior across lower levels since the packets are simply multiplied by peer degree as they are pulled towards lower levels. In practice, the following two issues can reduce the diffusion rate: (*i*) the independent packet scheduling by individual peers in level 1, may result in requesting duplicate packets from source, and (*ii*) the loss of delivered packets to level 1.

Source is the only node in the system that can keep track of delivered packets to each peer in level 1. Therefore, source can minimize the potential overlap among the delivered content to different diffusion sub-trees and maximize the diffusion rate. In PRIME, source implements two related mechanisms to achieve this goal as follows: First, it performs loss detection for delivered packets to each child peer and keeps track of the number of actually delivered copies for each packet. Second, any requested packet with timestamp ts that has already been delivered to other peers, is swapped with the rarest packet with the closest timestamp within a recent window $[ts - \Delta, ts]$ where $\Delta \gg RTT$. Performing loss detection ensures that the packet swapping mechanism effectively balances the number of copies of delivered packets.

V. Performance Evaluation

We use ns simulations to evaluate the effect of key design parameters on the performance of PRIME over a wide range of scenarios. Using packet level simulations has two important advantages compare to evaluation through experiments over a testbed such as PlanetLab as follows: (*i*) it enables us to investigate the effects of packet level dynamics (and packet loss) on system performance while capturing important details (e.g., location of losses at different parts of an overlay). (*ii*) it allows us to construct a wide range of evaluation scenarios by controlling key variables such as peer properties (e.g., level of bandwidth heterogeneity and asymmetry), resource availability and overlay connectivities.

A key challenge in the evaluation of PRIME is that changing a single parameter (e.g., source bandwidth) may have multiple related (and potentially conflicting) effects on system performance. A unique feature of our evaluation is to carefully untangle multiple effects of important parameters.

Simulation Setting: We use the following default settings in our simulations: the physical topology is generated with Brite [20] using 15 ASs with 10 routers per AS in top-down mode, the overlay is directed, the bandwidth-degree condition is satisfied, and the delay on each access link is randomly selected between [5 ms, 25 ms]. Core links have high bandwidth (ranging from 4 to 10 Gbps) and thus all connections experience bottlenecks only on the access links. Furthermore, all connections are congestion controlled using RAP [17], and all routers use RED queue management.

The delivered stream has 10 descriptions and all descriptions have the same constant bit rate of $\mathrm{C} = 160$ Kbps. Source performs loss detection and packet swapping. Each peer simulates the streaming consumption of delivered content after $\omega * \Delta$ seconds startup delay, and Δ is 6 seconds in all simulations.[3] Each simulation was run for 400 seconds. Our results represent the behavior of the system during the steady state after all peers have identified their parents and their pair-wise connections have reached their average bandwidth. Furthermore, our reported results are averaged across multiple runs of each scenario with different random seeds. We only focus on the *resource constraint* scenarios where supply is less than or equal to the demand for resources (i.e., bandwidth), i.e., resource index is less or equal to one. This allows us to stress test the protocol and ensures that the observed behavior is not a side effect of excess resources.

The following two scenarios are used as the *reference scenarios* in our evaluations: 200 homogeneous peers with (*i*) 700 Kbps and (*ii*) 1.5 Mbps access link bandwidth. Source bandwidth is set to the minimum value that ensures the delivery of sufficient stream quality $(peer_{bw}/C)$ to the overlay. In the first scenario source bandwidth is 800 Kbps and in the second it is 1.6 Mbps.

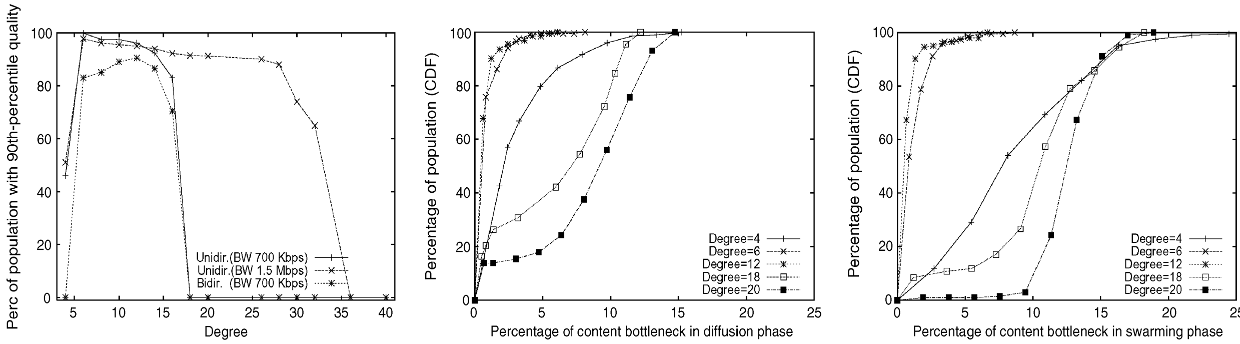
We also use the following methodology to decouple and separately quantify the impacts of bandwidth and content bottlenecks on delivered content from each parent. Each parent always sends packet to its children at the rate that is determined by a congestion controlled mechanism regardless of its useful content. At each packet transmission time to a particular child, if there is an outstanding list of requested packets from that child, the outgoing packet carries the first requested packet in the list. Otherwise, the parent sends an especially marked packet with the same size.

A. Peer Connectivity

Our goal is to answer the following question: "*How does the connectivity of individual peers (i.e., peer degree) affect the performance of content delivery in PRIME?*". Given a group of peers with certain bandwidth, increasing peer degree improves the connectivity among peers but reduces the value of bandwidth-per-flow (or $bwpf$) for each connection. Fig. 4(a) depicts the percentage of peers that receive at least 90% of the maximum deliverable quality (i.e., $inbw/C$) as a function of peer degree in the two reference scenarios. Note that for a fix population of peers, changing peer degree decreases the depth of the overlay. Therefore, for proper comparison, we adjust the value of ω based on the $depth$ of each overlay as follows: $\omega = depth + 3$. The number of swarming intervals is constant across these simulations $(K = 3)$. Fig. 4(a) shows two interesting points: (*i*) in each reference scenario, there is a sweet range of peer degree over which a majority of peers receive a high quality stream, (*ii*) the sweet range of peer degree has the same lower bound $(\mathrm{degree} = 6)$ in both scenarios but its upper bound depends on the bandwidth-degree ratio.

The poor performance of the system for small peer degrees (degree < 6) is due to the limited diversity of swarming parents

[3] We note that Δ does not have a significant impact on system performance as long as it is sufficiently larger than RTT. We have examined different values for Δ and selected 6 seconds as a representative value.



(a) (b) (c)

Fig. 4. (a) Percentage of peers that receive 90 percentile of the maximum quality across different degrees in uni- and bi-directional overlays. (b) and (c) Distribution of content bottleneck across different degrees in diffusion and swarming phases, respectively.

which leads to content bottleneck among peers. When peer degree is small, the number of diffusion sub-trees will be proportionally small because of the bandwidth-degree condition. This in turn proportionally reduces the probability that the randomly selected swarming parents for each peer would be located on different diffusion sub-trees and thus increases the probability of content bottleneck among peers regardless of peer bandwidth. The rapid drop in the delivered quality for large peer degrees is the result of significant increase in loss rate of individual connections. Fig. 4(a) clearly shows that the upper bound for the reference scenario with peer bandwidth 1.5 Mbps is almost twice as the the upper bound for peer bandwidth 700 Kbps. This demonstrates that the upper bound of the sweet range of peer degree is indeed a function of loss rate rather than the peer degree. We examine the effect of loss rate for higher peer degrees in further details later in this section.

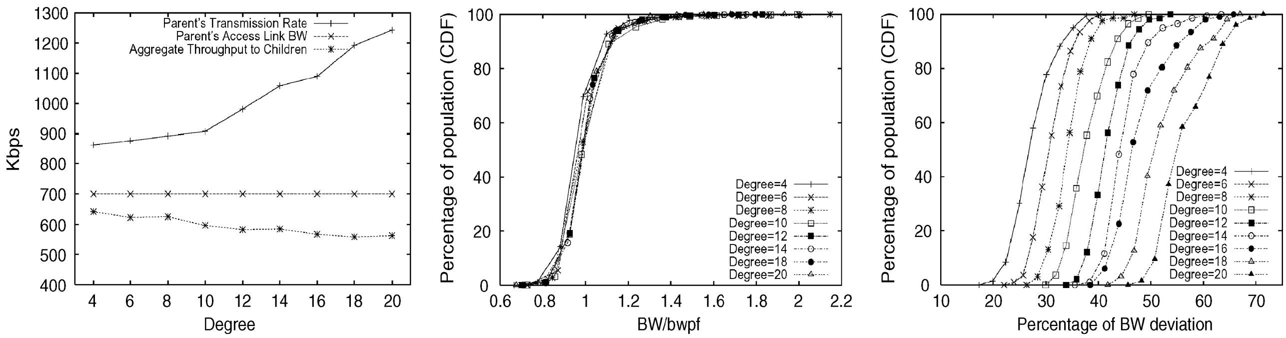
To verify our explanation, Figs. 4(b) and (c) depict the distribution of content bottlenecks from the diffusion and swarming parents among peers with peer bandwidth 700 Kbps for a few peer degrees, respectively. The percentage of content bottleneck from the diffusion (or swarming) parents is the percentage of congestion controlled bandwidth from the diffusion (or swarming) parent(s) that is not utilized for content delivery (i.e., the percentage of delivered packets that are especially marked). Comparing Figs. 4(b) and (c) shows that the percentage of content bottleneck is clearly higher from the swarming parents across all degrees which agrees with our discussion in Section IV-B. Furthermore, as we increase the peer degree from 4 to 6, the percentage of content bottleneck in both phases significantly decreases due to the improved diversity among swarming parents. However, any further increase in peer degree (beyond 12) reverses this trend and rapidly increases the percentage of content bottleneck in both phases due to the increase in loss rate.

Loss Rate: To further examine the effect of connection loss rate on system behavior for large peer degrees, Fig. 5(a) plots (from top to bottom) the aggregate transmission rate from a parent to all of its children, the parent's access link bandwidth and aggregate throughput to all of its children. The gap between the top two lines shows the bandwidth associated with lost packets at the outgoing access link of the parent peer whereas the gap between the bottom two lines represents the bandwidth associated with lost packets at the incoming access link of all children, collectively. This figure shows that the aggregate throughput from a parent peer to all of its children drops with increasing peer degree. More interestingly, while losses mostly occur at the parent's outgoing access link, a non-negligible fraction of losses also occur at the incoming access link of children as well. This suggests that throughput of some connections are limited by the parent's outgoing access link bandwidth while others are limited by the child's incoming access link bandwidth.

We further investigate the effect of loss rate by examining the distribution of normalized average throughput (normalized by the corresponding $bwpf$) and its deviation across all connections for different peer degrees in Fig. 5(b) and (c), respectively. These two figures paint an insightful picture on how the dynamics of congestion controlled bandwidth affect the location of bottleneck for individual connections. As peer degree increases, the distribution of normalized average throughput across all connections does not change but the distribution of its deviation shifts towards higher values. The larger deviation in per-connection bandwidth with larger peer degrees result in bottlenecks at both sender and receiver ends of individual connections. This in turn reduces the throughput of individual connections which causes bandwidth bottleneck for the corresponding child peers, and content bottleneck for all the descendant peers.[4]

Buffer Requirement: The poor performance outside the sweet range of peer degree indicates that the number of swarming intervals is inadequate for the delivery of the required number of data units to most of the peers due to the content bottleneck. This raises the following question: "*How many swarming intervals are required in a given scenario so that the majority of peers receive a high quality stream?*". Fig. 6(a) depicts the number of diffusion intervals (i.e., $depth$) and the minimum number of required swarming intervals ($K_{min} = \omega_{min} - depth$) as a function of peer degree in both reference scenarios (labeled as $K_{min} - Unidir$) such that 90% of peers receive 90% of the maximum deliverable quality. Fig. 6(a) shows that the depth of

[4]Conducting similar simulations with TFRC revealed that TFRC exhibits a lower loss rate but results in even lower utilization than RAP. In summary, when peer degree is large, an aggressive congestion control mechanism such as RAP may cause a rather higher loss rate and thus content bottleneck whereas slower congestion control mechanisms such as TFRC reduce the loss rate at the cost of lower utilization of resources.



(a) (b) (c)

Fig. 5. (a) Transmission rate of a selected peer along with its access link bandwidth and aggregate throughput to all of its children. (b) Distribution of $\mathrm{BW}/bwpf$ values across all connections. (c) Distribution of the deviation of aggregate bandwidth across all peers.

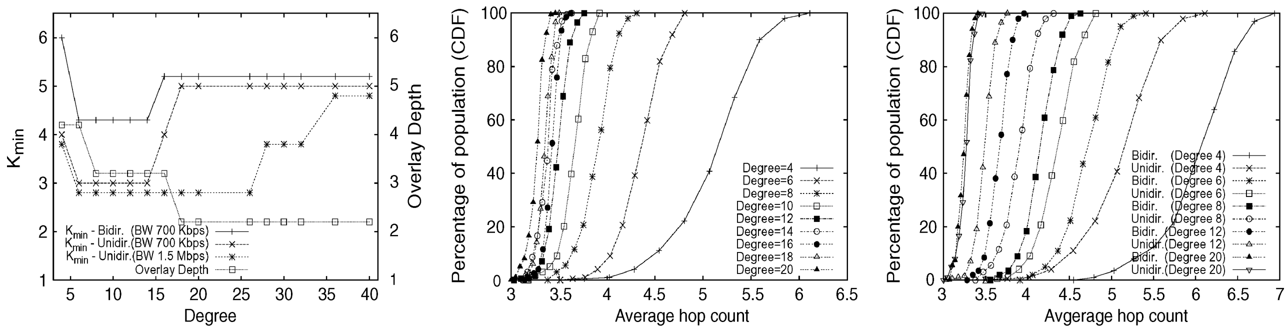
the overlay is independent of the peer bandwidth and gradually decreases with peer degree. As peer degree increases, K_{min} initially decreases from 4 to its minimum value of 3 intervals within the sweet range of peer degree due to the increasing diversity in the location of swarming parents across different diffusion subtrees. However, further increase of peer degree beyond a threshold results in the increase in K_{min} due to the higher loss rate and the resulting increase in content bottleneck which requires a longer swarming phase. In essence, this figure demonstrates (*i*) the minimum buffer requirement for individual peers (in a given scenario) in terms of the number of intervals as a function of peer degree (i.e., $\omega_{min} = depth + K_{min}$), and (*ii*) the direct relationship between K_{min} and $bwpf$ for different peer degrees.

Pattern of Content Delivery: We investigate the effect of peer degree on the pattern of content delivery by examining the following question "*How does the distribution of the average path length (in hops) among delivered packets to individual peers change as peer degree increases (i.e., the overlay becomes more connected)?*". Fig. 6(b) presents this distribution for average path length among peers for several peer degrees in the reference scenario with peer bandwidth 700 Kbps when the number of swarming intervals is equal to K_{min}. This figure reveals the following two important changes in the average path length to individual peers as overlay connectivity improves: (*i*) the average path length to individual peers monotonically decreases with peer degree primarily due to the decrease in overlay depth, (*ii*) the distribution of average path length among peers becomes more homogeneous. This is due to the increase in the diversity of swarming parents which in turn evens out the probability of content bottleneck among peers. The increasing homogeneity of average path length with peer degree also implies that lost packets are requested from the same parent during the following swarming interval(s) rather than through a longer path from other swarming parents.

Bi- vs. Uni-Directional Connectivity: Maintaining bi-directional connections between peers affects their connectivity. This raises the following question "*Is the performance of content delivery different over an undirected overlay (and if so, why)?*". To investigate this issue, we examine the reference scenario with 700 Kbps bandwidth but enforce bi-directional connections among peers. Fig. 4(a) shows the percentage of peers that receive 90% of the maximum deliverable quality over such a bidirectional overlay as a function of peer degree when K_{min} is 3 (labeled as Bidir.). This figure reveals that the percentage of peers with high quality in a bi-directional overlay is 10%–20% less than the uni-directional overlay over the sweet range of peer degree. Fig. 6(a) also shows the value of K_{min} for these bidirectional overlays as a function of peer degree. Fig. 6(a) indicates that bi-directional overlays require at least fsone extra swarming interval for peer degrees between 4 and 16. To explain this result, we note that bi-directional connections reduce the number of swarming shortcuts among diffusion sub-trees and thus increase the percentage of content bottleneck during the swarming phase. More specifically, for each diffusion connection from a parent to a child, there is a swarming connection in the reverse direction that connects two peers within the same diffusion sub-tree which is not an *effective* swarming shortcut. In a bidirectional overlay, effective swarming shortcuts between different sub-trees are established through connections between peers in the same level. Since most such "intra-level" connections are located at the bottom level, peers in higher levels of the overlay require a larger number of swarming intervals. Fig. 6(c) depicts the distribution of average path length for the above bidirectional overlays as well as the corresponding unidirectional overlays (that were shown in Fig. 6(b)) for easy comparison. This figure indicates that the distribution of average path length over the bi-directional overlay is around one hop longer than the uni-directional overlay for peer degree of 4. However, the difference in path lengths between bi- and uni-directional overlays rapidly diminishes with increasing peer degree. Note that the number of ineffective swarming shortcuts is roughly equal to the number of diffusion connections which is a function of the number of peers. Therefore, for a fixed population, as the peer degree increases, the extra connections must establish useful swarming shortcuts. This in turn improves the diversity of swarming parents and reduces the average hop count (and its deviations) for individual peers as shown in Fig. 6(c).

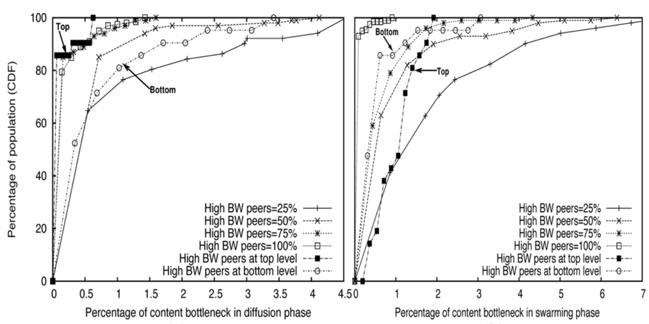
B. Bandwidth Heterogeneity

To investigate the effect of bandwidth heterogeneity, we consider the reference scenario with peer bandwidth 1.5 Mbps (bw_h) and reduce the access link bandwidth for a fraction of peers to bw_l. As we showed in Section III, the bandwidth-degree condition ensures a high utilization of access link among all peers even when peers have heterogeneous bandwidth. The percentage of content bottleneck for low bandwidth peers in heterogeneous scenarios is lower than homogeneous scenarios



(a) (b) (c)

Fig. 6. (a) K_{min}, and $depth$ in uni- and bi-directional overlays across different degrees. (b) Distribution of average path length across different peer degrees in uni-directional overlay with K_{min}. (c) Distribution of average path length across different degrees in uni- and bi-directional overlays with K_{min}.



(a) (b)

Fig. 7. Distribution of content bottleneck among high bandwidth peers in heterogeneous scenarios from diffusion (a) and swarm (b) parents.

since some of their swarming parents are likely to be high bandwidth peers with higher available quality. Therefore, we focus on the delivered quality to high bandwidth peers. The first question is: "*How are the delivered quality and buffer requirement of high bandwidth peers affected by the percentage of low bandwidth peers?*".

Figs. 7(a) and (b) show the distribution of content bottleneck among high bandwidth peers ($bw = 1.5$ Mbps) with different percentage of low bandwidth peers (1 Mbps) from diffusion and swarming parents, respectively. These figures show that the percentage of high bandwidth peers has a minor impact on the content bottleneck in both phases. Figs. 7(a) and (b) show a minor increase in content bottleneck from the diffusion and swarming parents when the percentage of high bandwidth peers is small. In the diffusion phase, this is due to the decrease in the total number of overlay connections and the resulting increase in the overlay depth. In the swarming phase, the percentage of content bottleneck at each peer depends on the aggregate available content among its swarming parents. As the number of high bandwidth peers decreases, a larger fraction of their swarming parents are likely to be low bandwidth peers. This in turn reduces the aggregate available quality among their swarming parents and increases the probability of content bottleneck among high bandwidth peers. We have also examined other scenarios with different levels of bandwidth heterogeneity (bw_h/bw_l) and observed that the level of heterogeneity does not have any impact on the delivered quality to high bandwidth peers.

Location of High Bandwidth Peers: Another important question in an overlay with heterogeneous peers is: "*How does the location of high bandwidth peers in the overlay affect the percentage of content bottleneck among them?*". To examine this issue, we explore a heterogeneous scenario where only 10% of peers have access link bandwidth of 1.5 Mbps and the remaining peers have access link bandwidth of 1 Mbps. We enforce the overlay construction mechanism to only place high bandwidth peers at the top level (as source's children) or at the bottom level. Figs. 7(a) and (b) show the percentage of content bottleneck for these two cases (labeled as "top" and "bottom") for comparison with previous scenarios. Placing the high bandwidth peers in non-bottom levels reduces the depth of the overlay and thus reduces the required number of diffusion intervals. However, it also reduces the connectivity among the diffusion sub-trees and thus increases the probability of content bottleneck from the swarming parents. In contrast, placing high bandwidth peers at the bottom level slightly increases overlay depth and thus increases the content bottleneck in diffusion phase. However, this effect is compensated by the higher connectivity among the diffusion sub-trees which decreases the probability of content bottleneck from the swarming parents. *In summary, the location of high bandwidth peers in the overlay has an opposite effect on the probability of content bottleneck in diffusion and swarming phases. Therefore, the overall impact on the performance of content delivery and the minimum buffer requirement* (i.e., ω) *is relatively small.*

C. Packet Scheduling

In Section IV-C, we presented the criteria for description selection (i.e., random, rarest) and parent selection (i.e., least proportionally loaded, random) and the relative order of selection (between description and parent) as basic design choices for packet scheduling scheme. These choices lead to six variants of the packet scheduling scheme. In this subsection, we compare the performance of these six variants of the scheduling in the reference scenario with access link bandwidth of 700 Kbps and assume that all peers use the same packet scheduling scheme. Fig. 8(a) depicts the percentage of peers that receive 90% of the maximum deliverable quality as a function of peer degree for these six packet scheduling schemes where $\omega = depth + 3$. This figure illustrates two interesting points: First, except for the two scheduling schemes that randomly select the parent, the performance of other schemes is very similar within the sweet range of peer degree. *This suggests that neither the criteria for selecting the description of a packet nor the relative order of selection (between description and parent) significantly affects the performance of packet scheduling schemes*. Second,

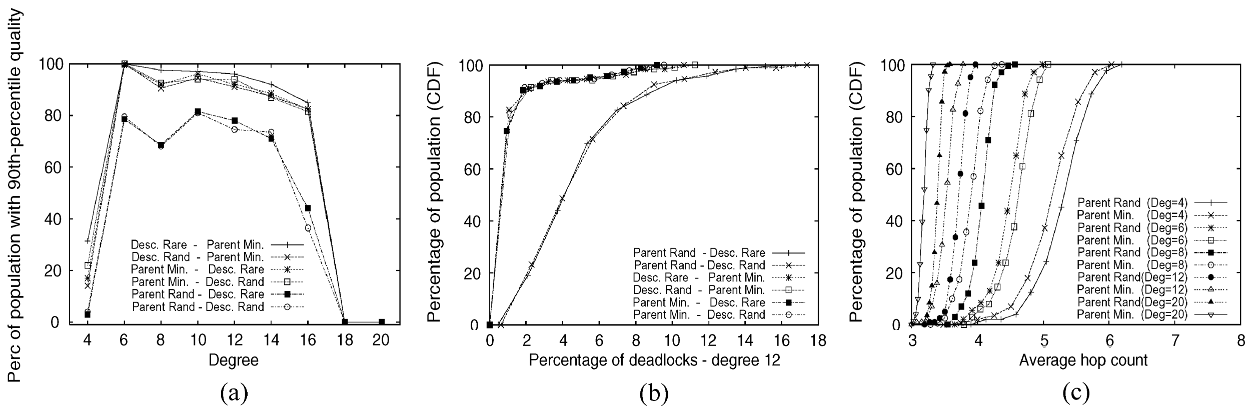


Fig. 8. (a) Percentage of peers with 90 percentile delivered quality for different peer degrees with various scheduling algorithms. (b) Distribution of the frequency of deadlocks for peer degree of 12 across various scheduling algorithms. (c) Distribution of average path length for two high and low performing scheduling algorithms across different degrees.

the percentage of peers that receive a high quality stream in the two low-performing schemes (labeled as $ParentRand - Desc.\ Rand/Rare$) is very similar, and roughly 20% lower than other schemes within the sweet range of peer degree.

Intuitively, those schedulings which request a packet from a random parent are more likely to experience content bottleneck due to the higher frequency of *deadlock*s during parent selection. A deadlock event occurs when a required packet is available among some parents but it can not be requested since the bandwidth budget of those parents are fully allocated for delivery of other packets. To verify this hypothesis, Fig. 8(b) depicts the distribution of frequency of deadlock (i.e., the fraction of packets that experience deadlock during the scheduling process) among peers for all six schedulings when peer degree is 12. Fig. 8(b) clearly shows that the median frequency of deadlock is roughly four times higher for schedulings that use random parent selection. The random parent selection may not request all the unique packets from individual parents. Therefore, a fraction of bandwidth budget from diffusion parents is used for the delivery of packets that are already available at other parents.

Closer examination of the two low-performing scheduling schemes reveals that these two schemes can achieve good performance with an extra swarming interval (i.e., larger buffer, ω). This raises the following interesting question "*Does extra swarming intervals accommodate the delivery of deadlocked packets through longer paths to reduce the frequency of deadlock?*". Fig. 8(c) depicts the distribution of average path length (in hops) across delivered packets for one of the high-performing scheduling scheme ($ParentMin. - Desc.Rand$) as a reference and one of the low-performing scheduling scheme ($ParentRand - Desc.Rand$) (with a proper number of swarming intervals) across different peer degrees. Fig. 8(c) reveals that the average path length for the low-performing scheduling scheme with longer swarming is around 20% longer for all peer degrees. This suggests that 20% of peers that have poor performance in Fig. 8(a), can leverage the extra swarming interval to request the deadlocked packets from another swarming parent. The larger number of swarming intervals increases the pool of swarming packets and decreases the probability of deadlock event.

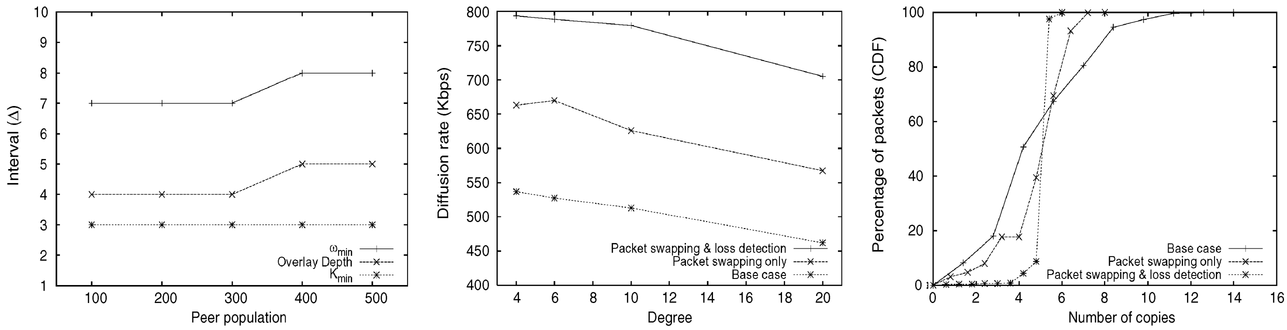
D. Peer Population

We examine the scalability of PRIME protocol by addressing the following question: "*How do the delivered quality and buffer requirement at individual peers change with peer population?*". Fig. 9(a) shows the duration of diffusion phase (or overlay $depth$), the minimum duration of swarming phase (K_{min}) and the minimum buffer requirement (or ω_{min}) as a function of peer population in the reference scenario with access link bandwidth of 700 Kbps when peer degree is 6. This figure provides a good evidence of the scalability of PRIME with user population. As the peer population increases, overlay depth slowly grows but the duration of the swarming phase (with a proper peer degree) remains constant. To explain this, we note that increasing peer population does not affect the number of diffusion sub-trees. This means that the diversity of swarming parents for individual peers does not change with peer population. Therefore, the observed content bottleneck and the required number of swarming intervals for individual peers does not change with peer population. We have observed the same behavior for different degrees within the sweet range of peer degree. *The observed trend in this result suggests that within the sweet range of peer degree, PRIME can effectively utilize available resources in the system and provide maximum quality to peers in a scalable fashion if the buffer size is logarithmically increased with peer population.*

E. Source Behavior

In this subsection, we examine the effect of the following two orthogonal aspects of source behavior on the system performance: (*i*) Packet swapping and loss detection, and (*ii*) Source bandwidth.

Packet Swapping & Loss Detection: We explore the effect of source coordination in the reference scenario with 700 Kbps access link bandwidth where source bandwidth and K_{min} are 800 Kbps and 3, respectively. This configuration ensures all peers in level 1 receive a high quality stream. Fig. 9(b) depicts the delivered quality from source to level 1 (i.e., diffusion rate to level 1) as a function of peer degree in three different scenarios: (*i*) source without any coordination, (*ii*) source with only packet swapping, and (*iii*) source with both packet



(a) (b) (c)

Fig. 9. (a) K_{min}, $depth$ and ω_{min} for different peer populations. (b) Diffusion rate to level 1 across various peer degrees in three scenarios: source without any coordination, source with only packet swapping and source with both packet swapping and loss detection. (c) Distribution of the number of copies of each packet that are delivered to level 1 for peer degree 10 in the three scenarios.

swapping and loss detection. Note that the outgoing bandwidth from source is fully utilized across these scenarios and its aggregate throughput to level 1 is not affected by the coordination mechanism. Fig. 9(b) shows that the diffusion rate slowly decreases with peer degree in all three scenarios due to the increase in loss rate (as we described in Fig. 5(a)). Incorporating packet swapping significantly increases the diffusion rate, and adding loss detection leads to further improvement in the diffusion rate. Fig. 9(c) depicts the distribution of the number of delivered copies for individual packets to level 1 in the above three scenarios when peer degree is 10. This figure clearly illustrates that incorporating packet swapping and then loss detection progressively balances out the number of copies of delivered packets to level 1. *In summary, incorporating packet swapping and loss detection enable us to deliver certain quality with a lower source bandwidth or to improve delivered quality for a given source bandwidth.*

Source Bandwidth: Another key question is "*How does excess source bandwidth affect delivered quality and buffer requirement at individual peers?*". Fig. 10(a) shows the effect of excess source bandwidth (beyond the required stream bandwidth of 700 Kbps) on the following properties in the reference scenario with peer access link bandwidth of 700 Kbps and degree of 6: (*i*) aggregate throughput to level 1 (labeled as Data Rate), (*ii*) diffusion rate, (*iii*) overlay $depth$ and (*iv*) total buffering at each peer (ω). The x axis represents the normalized value of excess source bandwidth, i.e., $(SourceBW - 700$ Kbps$)/700$ Kbps. Fig. 10(a) illustrates that increasing source bandwidth has two effects: First, it increases the source degree because of the bandwidth-degree condition. This slowly reduces the overlay depth and thus decreases the buffer requirement at individual peers. Second, increasing source bandwidth (with packet swapping and loss detection) initially increases the number of diffusion sub-trees with unique content and thus improves the diffusion rate until it reaches the maximum available quality at the source. Given a fixed peer population, increasing the number of unique diffusion sub-trees decreases the population of peers in each sub-tree which results in a lower probability of having an inefficient swarming connection (intra-subtree swarming shortcuts). Therefore, increasing source bandwidth reduces the percentage of content bottleneck from the swarming parents as shown in Fig. 10(b). Once the delivered quality to level 1 is saturated, any further increase of source bandwidth results in adding redundant diffusion sub-trees (that do not have unique content). This reduces overlay depth and slightly reduces content bottleneck during the diffusion phase.

F. Peer Dynamics

So far we have not considered the effect of peer dynamics (or churn) in our simulations. In practice, churn may have both short-term (or transient) and long-term effects on the performance of content delivery in PRIME. When a peer leaves the overlay, the aggregate bandwidth to its children is dropped until each child manages to establish a connection to a new parent. The transient effect of parent departure on delivered quality to a child depends on the efficiency of the parent discovery (i.e., time to connect to a new parent) and the amount of buffered content at the child among other things. Over a longer term, churn could change the bandwidth-to-degree ratio among peers in the overlay. We call such an overlay a *distorted* overlay where the bandwidth-to-degree condition is not satisfied. We focus on this long-term effect of churn on the performance of content delivery since it is more significant than the transient effect and it does not depend on protocol-specific details.

To examine the performance of content delivery over a distorted overlay, we consider the reference scenario with peer access link bandwidth of 700 Kbps, peer degree 6 and $\omega = depth + 3$ where bandwidth-degree condition is satisfied. We emulate a distorted overlay by removing $ch\%$ of randomly selected peers from the reference scenario without allowing remaining peers to establish new connections. We can control the level of distortion by changing the percentage of departed peers (ch). The resulting distorted overlay represents the snapshot of the overlay structure as peers join and leave the system. As the level of distortion increases, the distribution of peer population across different levels of the overlay becomes more imbalanced compare to a properly connected overlay and the depth of the overlay may increase.

Fig. 10(c) depicts the distribution of average delivered quality among peers for different levels of distortion. This figure reveals that the delivered quality to peers is rather sensitive to the level of distortion and rapidly drops as ch passes 30%. One key question is "*Is the decrease in delivered quality due to the drop in the utilization of access link bandwidth (i.e., bandwidth bottleneck) or the inability of peers to utilize the available bandwidth (i.e., content bottleneck)?*". Fig. 11(a) shows the distribution of incoming access link utilization among peers for dif-

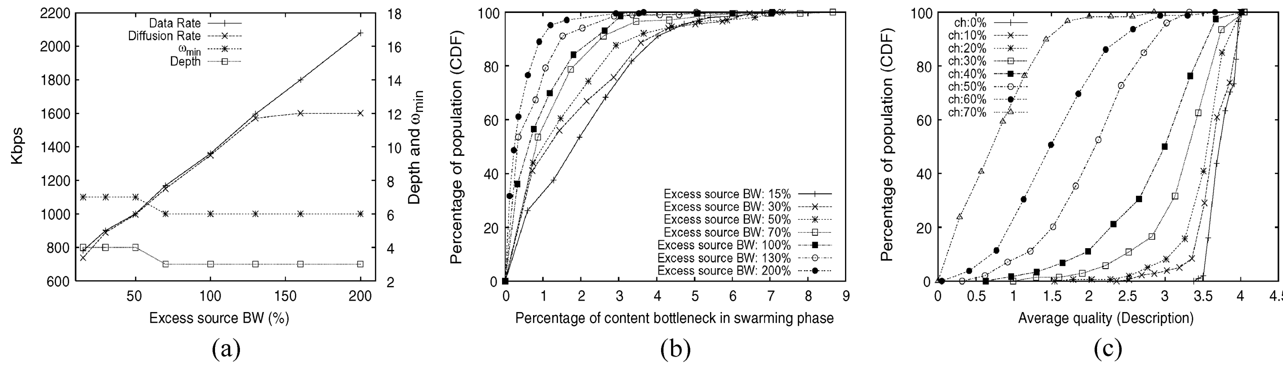


Fig. 10. (a) Diffusion rate and data rate to level 1 along with ω and depth across various excess source bandwidth. (b) Distribution of content bottleneck across various excess source bandwidth from the swarming parents. (c) Distribution of average delivered quality (descriptions) to each peer for various percentages of ch.

ferent levels of distortion. This figure indicates that the utilization of access link bandwidth drops with the number of departed peers. However, comparing Figs. 10(c) and 11(a) illustrates that the decrease in delivered quality is visibly larger than the drop in access link utilization when level of distortion in the overlay is roughly larger than 30%. This suggests that both bandwidth and content bottleneck contribute into the drop in quality as the overlay becomes more distorted.

To identify the underlying causes for content bottleneck in distorted overlays, we examine average diffusion rate at each level of the overlay as distortion increases in Fig. 11(b). Fig. 11(b) demonstrates that the diffusion rate at the top level is not affected by the percentage of departed peers as long as the number of peers in level 1 is not affected. However, the diffusion rate at all lower levels is rapidly dropped once more than 30% of peers depart. A closer examination of the overlay connectivity revealed that when a large fraction of peers depart, some diffusion sub-trees may become disconnected (especially at the higher levels) from the rest of the overlay, e.g., a peer in level 1 does not have any child. Such an event has a ripple effect and reduces the diffusion rate to all the lower levels of the overlay due to the content bottleneck. This implies that increasing the number of swarming intervals does not improve delivered quality in these scenarios. We have conducted simulations with longer buffer sizes and confirmed this observation. *In summary, as the overlay becomes more distorted, the delivered quality to individual peers is dropped due to both bandwidth and content bottleneck. The content bottleneck is caused by the disconnection of some diffusion sub-trees from the rest of the overlay.*

G. Limited Resources

In all the previous subsections, we assumed that participating peers have symmetric access link bandwidth and their downlink bandwidth is equal to the stream bandwidth. In such a scenario, the aggregate demand and supply for bandwidth are equal, and the ratio of demand to the supply for bandwidth which is called *resource index* (RI), is one. In practice, the uplink bandwidth that a peer is able or willing to contribute might be less than its incoming bandwidth. Therefore, the aggregate resources may not be sufficient to provide maximum deliverable quality to all peers. In such a resource-constraint scenario, the key question is "*Is the drop in quality fairly similar across participating peers?*".

Fig. 11(c) depicts the distribution of delivered quality among all peers in the reference scenario with incoming peer bandwidth 700 Kbps, peer degree 6 and consistent bandwidth-degree ratio among peers for resource index values 1.0, 0.8, and 0.6. This figure shows that the distribution of delivered quality is relatively skewed among peers. Decreasing resource index reduces the average delivered quality among peers, and the shape of its distribution becomes slightly more skewed. When the amount of aggregate outgoing bandwidth in the system is insufficient, a random subset of peers are unable to establish connection to the adequate number of parents and thus receive lower quality. The extent of the observed deficit in resources is variable among peers which leads to a skewed distribution of delivered quality among them.

H. Presence of Free-Riders

A key challenge in any P2P system is to gracefully accommodate (or at least limit the potential damage by) uncooperative peers that do not contribute any resource (i.e., free-riders). We examine the effect of free-riders on PRIME performance in the reference scenario with incoming peer bandwidth 700 Kbps, and peer degree 6. We focus on a scenario when resource index is one to investigate any direct impact of free-riders on performance[5] (as opposed to the effect of insufficient resources).

Fig. 11(c) also depicts the distribution of delivered quality among peers when 10% and 50% of peers are free-riders. This figure reveals that as long as there is sufficient resource in the system, the presence of free-riders should not have any significant effect on the performance of content delivery. The exception is the scenario when free riders disconnect a particular diffusion sub-tree from the rest of the overlay. In such an event, the data units that are delivered through the disconnected diffusion sub-tree(s) can not reach other peers in the overlay during the swarming phase and thus delivered quality to the rest of the overlay is proportionally dropped. Fig. 11(c) presents such a scenario (the line labeled as RI:1-Fr:50%-Heter) where 50% of peers are free-riders and the rest have heterogeneous outgoing bandwidth (25% 240 Kbps, 25% 2.5 Mbps). In essence, presence of a group of free-riders affects the connectivity among sub-trees and the performance of content delivery even if available resources are sufficient. Clearly, the larger the percentage of free-riders or the closer their distance from the source, the

[5]To maintain the resource index at one in the presence of free riders, cooperative peers have to contribute more bandwidth than they use.

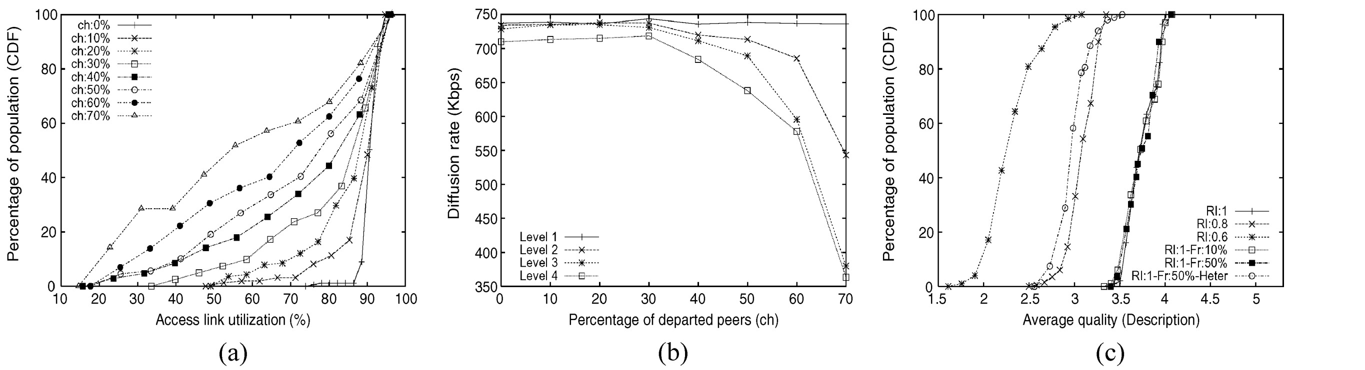


Fig. 11. (a) Distribution of utilization of access link bandwidth among peers for various percentages of ch. (b) Diffusion rate to different levels for various percentages of ch. (c) Distribution of average delivered quality (layers) to each peer for various RIs and for scenarios with different percentage of free-riders with $\mathrm{RI} = 1$. The result for 50% free-riders in a heterogeneous scenario is also included (RI:1-Fr:50%-Heter).

more likely that a diffusion sub-tree becomes disconnected. In our future work, we plan to design light weight techniques to detect free-riders in the overlay and minimize their adverse impact on content delivery.

VI. Conclusions and Future Work

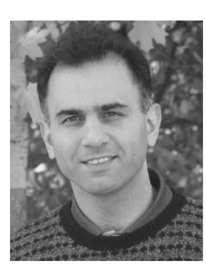
In this paper, we presented PRIME, a mesh-based P2P streaming mechanism for live content that can effectively incorporate swarming content delivery. We argued that the bandwidth-degree condition should be satisfied by the overlay construction mechanism in order to minimize the bandwidth bottleneck among participating peers. We also derived the pattern of content delivery that can incorporate swarming in order to effectively utilize the outgoing bandwidth of participating peers and thus minimize the content bottleneck in the system. This in turn led us to the desired packet scheduling scheme at individual peers. Through extensive ns simulations, we examined the effect of key factors on PRIME performance and identified a few fundamental design tradeoffs.

We are currently extending this work along several dimensions. First, we are examining the effect of ongoing churn on PRIME performance, in particular on ensuring the bandwidth-degree condition. Second, we are evaluating PRIME performance in scenarios where the distribution of outgoing bandwidth is very skewed or in the presence of free-riders [21]. Third, we also use PRIME to conduct systematic comparison between tree-based and mesh-based P2P streaming mechanism [2]. Fourth, we have prototyped PRIME and currently conducting experiments over PlanetLab. Finally, we plan to incorporate the notion of "contribution awareness" into PRIME to cope with uncooperative users.

References

[1] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *ICNP*, 2003.
[2] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of P2P live streaming services," in *INFOCOM*, 2007.
[3] B. Cohen, Bittorrent. [Online]. Available: http://www.bittorrent.com
[4] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: High bandwidth data dissemination using an overlay mesh," in *SOSP*, 2003.
[5] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, "Chainsaw: Eliminating trees from overlay multicast," in *IPTPS*, 2005.
[6] A. Vlavianos, M. Iliofotou, and M. Faloutsos, "Bitos; Enhancing bittorrent for supporting streaming applications," in *GI*, 2006.
[7] W. T. Ooi, "Dagster: Contributor aware end-host multicast for media streaming in heterogeneous environment," in *SPIE MMCN*, Jan. 2005.
[8] C. Dana, D. Li, D. Harrison, and C. Chuah, "Bass: Bittorrent assisted streaming system for video-on-demand," in *IEEE MMSP*, 2005.
[9] F. Pianese, J. Keller, and E. W. Biersack, "Pulse, a flexible P2P live streaming system," in *GI*, 2006.
[10] X. Liao *et al.*, "Anysee: Scalable live streaming service based on inter-overlay optimization," in *INFOCOM*, 2006.
[11] M. Castro, P. Druschel, A.-M. Kermarrec, A. R. A. Nandi, and A. Singh, "SplitStream: High-bandwidth content distribution in a cooperative environment," in *ACM SOSP*, 2003.
[12] V. Venkataraman, K. Yoshida, and P. Francis, "Chunkyspread: Heterogeneous unstructured end system multicast," in *ICNP*, 2006.
[13] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "Coolstreaming: A data-driven overlay network for live media streaming," in *INFOCOM*, 2005.
[14] S. Xie, G. Y. Keung, and B. Li, "Measurement of a large-scale peer-to-peer live video streaming system," in *ICPPW*, 2007.
[15] S. Xie, B. Li, G. Keung, and X. Zhang, "Large Scale Peer-to-Peer Live Video Streaming: Theory and Practice," 2006, Tech. Rep..
[16] Y. Choe, D. Schuff, J. Dyaberi, and V. Pai, "Improving vod server efficiency with BitTorrent," *ACM Multimedia*, 2007.
[17] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *INFOCOM*, 1999.
[18] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for Unicaqt applications," in *ACM SIGCOMM*, 2000.
[19] N. Magharei and R. Rejaie, "Understanding mesh-based peer-to-peer streaming," in *NOSSDAV*, 2006.
[20] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *MASCOTS*, 2001.
[21] N. Magharei, Y. Guo, and R. Rejaie, "Issues in offering live P2P streaming service to residential users," in *IEEE CCNC*, 2007.

Nazanin Magharei received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2002. She is currently pursuing the Ph.D. degree in the computer science at the University of Oregon, Eugene. Her research interests include P2P streaming, P2P overlay characterization and network measurement.



Reza Rejaie (SM'06) received the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1996 and 1999, respectively, and the B.S. degree from the Sharif University of Technology, Iran, in 1991.

He is currently an Associate Professor at the University of Oregon, Eugene. From 1999 to 2002, he was a Senior Technical Staff Member at AT&T Labs—Research in Menlo Park, CA. His research interests incldue P2P networking, network measurement and multimedia networking.

Dr. Rejaie received an NSF CAREER Award for his work on P2P streaming in 2005. He has been a Senior Member of the Association for Computing Machinery (ACM) since 2006.