Dissecting the performance of Live Mesh-based Peer-to-Peer Streaming

Nazanin Magharei, Reza Rejaie
Email: {nazanin,reza}@cs.uoregon.edu

***Abstract*—Mesh-based Peer-to-Peer streaming (MPS) mechanisms incorporate swarming content delivery and thus are able to support scalable streaming of live content. Their key component is a packet scheduling scheme at each peer that determines pulled packet from neighbors while accommodating in-time arrival and diversity of delivered packets. Besides packet scheduling scheme, the overall performance of a MPS mechanism also depends on the availability of excess resources in the system. Recently proposed MPS mechanisms have been evaluated in a scenario-rich setting. Thus, neither their performance in a resource constraint scenario nor the separate effect of packet scheduling and available resources on their performance is known.**

In this paper, we dissect the performance of MPS mechanism and investigate the effect of packet scheduling and available resource on their performance. We argue that the global pattern of content delivery primarily determines behavior of a MPS system. We present a new evaluation methodology that properly captures this pattern. Using our evaluation methodology, we examine the performance of representative scheduling schemes and the role of available resources. Our findings provide useful insights in design and evaluation of MPS mechanisms.

I. Introduction

Mesh-based Peer-to-Peer streaming (MPS) mechanisms offer a promising approach for scalable streaming of live content over the Internet. in MPS participating peers form a randomly connected mesh over which they incorporate swarming (*i.e.*, pull-based) content delivery. Swarming content delivery enables participating peers to contribute their resources (*i.e.*, outgoing bandwidth) more effectively which in turn improves the utilization of available resources among peers, and leads to a better scaling property for MPS approach compared to the traditional tree-based Peer-to-Peer streaming approach [1]. Incorporating swarming content delivery into P2P streaming is inspired by the success of swarming for delivery of static files (*e.g.*, BitTorrent [2]). File swarming mechanisms leverage the availability of the entire content and provide different segments of the content to participating peers. This enables participating peers to exchange their available segments and effectively contribute their outgoing bandwidth. The key component of swarming content delivery is a *packet scheduling* scheme at individual peers which determines the subset of packets that should be pulled from each neighbors.

Incorporating swarming content delivery into MPS of *live* content is inherently challenging because it should accommodate the following two requirements: *(i)*: the in-time delivery of each packet to individual peers (*i.e.*, timing requirement), and *(ii)* the diversity of available packets among peers in order to enable effective swarming (*i.e.*, diversity requirement). These are potentially conflicting requirements for packet scheduling because addressing the timing requirement demands for pulling missing packets with earlier playout time whereas addressing the diversity requirement demands for pulling missing packets in a random (or rarest-first) fashion. In essence, swarming content delivery in MPS should be "timing-aware" and properly leverage the conflicting demand between timing and diversity. Another difficulty is that the source constantly generates new packets in live streaming sessions. Therefore, the pool of newly available packets for delivery is very small (compared to the entire content in file swarming) which could in turn limit the degree of diversity in available packets among peers and adversely affect swarming content delivery. This implies that a packet scheduling scheme for a MPS mechanisms of live content should be carefully designed to address this conflicting requirements. However, the availability of excess resources (*i.e.*, source and peer bandwidth) or large buffer at each peer could relax the timing requirement (thus the need for a well designed packet scheduling scheme) and still deliver high quality stream to individual peers.

A few recent studies have proposed new MPS mechanisms that incorporates a variety of scheduling schemes ranging from simple pull-rarest-first [3] to prioritizing packets based on various combinations of their playtime and rarity [4]. These studies often evaluate their proposed mechanisms through simulations in a resource-rich scenario [5] or through actual deployment where available resources (especially peer bandwidth) are not accurately known. Despite the challenges in accommodating the conflicting requirements for packet scheduling, these studies have all reported high delivered quality to participating peers. This raises the following important question: *"Does the reported performance in previous studies on MPS mechanisms represent the intrinsic ability of their scheduling scheme to utilize available resources or it is merely the side effect of abundant resources in their evaluation?"*. In a nutshell, the following two important issues about proposed MPS mechanisms of live content have not been adequately addressed by previous studies:

- *Effect of Packet Scheduling Scheme*: How does a packet scheduling scheme perform in a scenario with limited resources? What aspects of packet scheduling scheme primarily result in the observed (good or bad) performance?
- *Effect of Available Resources*: How does the availability of various types of excess resources (*i.e.*, peer and source bandwidth) affect the performance of a packet scheduling scheme? What are the underlying causes for the observed effect of excess resources?
- *Effect of Buffering*: How does the available buffer size at each peer interact with the packet scheduling scheme at individual peers and available resources?

In this paper, we dissect the performance of MPS mechanisms of live content to systematically examine the impact of both packet scheduling scheme and available resources on their overall performance. Our key observation is that the performance of a packet scheduling scheme primarily depends on the global pattern of content delivery from source to individual peers through the overlay. Therefore, capturing the global pattern of content delivery is very useful in identifying any potential performance bottleneck for content delivery. We present an evaluation methodology to properly capture the global pattern of content delivery, and examine its characteristics. We also derive the proper pattern of content delivery that minimizes the required resources and buffer-size in the system, and present a set of characteristics to identify such a pattern, *i.e.*, the signature of a proper pattern of delivery. The proper pattern of content delivery then serves as a reference to identify the underlying causes in a poorly performing scheduling scheme (*i.e.*, performance bottlenecks).

We identify the design space of the packet scheduling schemes by exploring different ways to address the conflicting requirement between timing and diversity in timing aware swarming mechanisms. Then, we select several candidate scheduling schemes that represent the entire design space as well as the key features in the previously proposed scheduling schemes. Using our evaluation methodology, we examine the performance of each candidate schemes in both resource constraint and resource-rich environments. This illustrates the ability of our methodology to assess the separate effect of scheduling scheme and available resources on the performance of MPS mechanisms. Overall, our study provides a useful insight in the design of packet scheduling scheme by revealing how different components of scheduling affect the global pattern of content delivery. Our evaluation methodology in essence offers a unified framework for head-to-head comparison of different packet scheduling schemes. Furthermore, our findings provide useful guidelines for resource provisioning and stress testing of MPS systems. Our main findings can be summarized as follows:

- Scheduling schemes that prioritize newly available packets with largest timestamps among parents exhibit a significantly better performance than other schemes. Only this class of scheduling schemes can achieve good performance in resource constraint scenario. This implies that in a timing-aware swarming content delivery the availability of new packets is more important than addressing timing requirement.
- Increasing source bandwidth (with proper source coordination) results in a major improvement in performance compared to increasing peer bandwidth. This is due to the unique role of source bandwidth on the rate and timing of delivered packets to participating peers throughout the overlay. In contrast increasing peer bandwidth has a limited effect on performance.
- Any poorly designed scheduling can provide good quality to participating peers by adding sufficient amount of excess resources of proper type and/or increasing buffer size.
- Our derived signature/condition for good pattern of content delivery properly represents the characteristics of a well-behaved scheduling scheme.

The rest of this paper is organized as follows: Section II presents the required background on MPS of live content for this paper. In Section III, we explore the design space for the packet scheduling scheme for timing-aware swarming, and identify several candidate schemes. We describe our evaluation methodology in Section IV. In Section V and VI, we discuss the effect of scheduling scheme and available resources on the performance of MPS, respectively. Finally, Section VIII concludes the paper and sketches our future plans.

II. Mesh-based P2P Streaming: Background

In MPS, participating peers form a randomly connected mesh (*i.e.*, unstructured overlay) over which they incorporate swarming content delivery. To consider the general case, in this paper we assume that peers form a directed overlay where there is a parent-child relationship between peers. [1] Each peer learns about a random subset of participating peers from a bootstrapping node and tries to maintain connection to a proper number of parent peers while limiting the number of its child peers. The number of parents and children for each peer are proportional with its incoming and outgoing access link bandwidth, respectively. This balances out the load among peers and thus minimizes the possibility of major bottleneck on the access link bandwidth.

Swarming content delivery is a key component of MPS. We assume that all data connections between parent peers and their children are congestion controlled using RAP or TFRC. Each peer (as a parent) periodically reports its available content to its child peers. The packet scheduling scheme at each peer (as a child) periodically (once per Δ second) determines a subset of packets that should be requested (*i.e.*, pulled) from each parent. The collective behavior of packet scheduling scheme at all peers determines the global pattern of pull content delivery from source to individual peers though the overlay. This global pattern of delivery directly affects (i) the availability and thus arrival time of packets to individual peers, and (ii) the diversity of available packets among connected peers.

In the context of live MPS sessions, all participating peers maintain a loosely synchronized playout time that is ω seconds behind source playout time (Figure 1). This implies that each peer requires to maintain at least ω seconds worth of buffering to absorb out-of-order delivery of packets that are caused by swarming. Maintaining a close playout time maximizes the overlap between relevant packets among participating peers. This not only facilitates packet swapping among peers but also greatly simplifies parent selection. [2] In essence, at any point

[1] An undirected overlay is indeed a special case [6] of directed overlay, and thus most of our discussion and findings are still valid.

[2] If participating peers maintain different playout time, each peer can only select a parent with overlap in ints relevant packets which limits the number of peers that can serve as its parent.

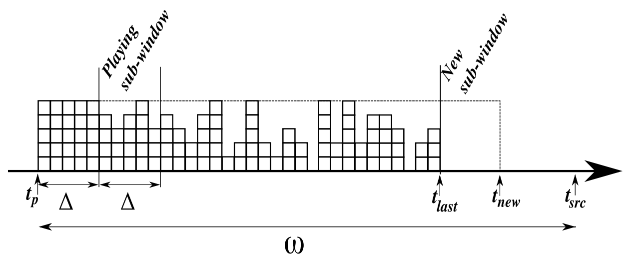


Fig. 1. Buffer state at an scheduling event in a peer

of time, participating peers swarm the most recent ω seconds window of content that source has generated.

The performance of a MPS mechanism of live content depends on various design choices such as packet scheduling scheme, peer connectivity, source behavior and buffer size, as well as available resources in the system namely source bandwidth and peer bandwidth. For example, increasing source or peer bandwidth, or providing larger buffer to individual peer intuitively increases the chance for in time delivery of packets and thus improves delivered quality to individual peers.

Previously proposed MPS mechanisms have usually been evaluated in a particular scenario with either unknown or abundant amount of resources. Therefore, it is difficult to assess whether their reported good performance is a side effect of excess resources or an inherent ability of the proposed scheduling scheme to utilize available resources. Our goal in this paper is to address this issue and illustrate the role of packet scheduling scheme and available resources on the performance of MPS systems of live content.

III. Packet Scheduling: design space and candidate schemes

In this section, we identify the design space for packet scheduling schemes in MPS of live content, and then select a few candidate schemes that properly represent interesting scheduling schemes across the space. The packet scheduling at each peer should determine the requested packets from each parent based on the following information *(i)* the missing packets that still have sufficient time to be pulled, *(ii)* available packets among parents based on their reports, and *(iii)* congestion controlled bandwidth from each parent that is passively measured by each child. The total number of requested packets from all parents during one interval (*i.e.*, packet budget) is determined by the aggregate bandwidth from all parents, while the number of pulled packets from each parent is proportional to its contributed bandwidth.

The goal of the scheduling scheme at each peer is to ensure in-time delivery of requested packets while addressing the diversity of available packets among peers. The packet scheduling function is invoked once per Δ seconds and in each scheduling event it considers packets within its current window of ω seconds (buffer) that should be pulled from parents in the current interval. The timestamp of the packets in the current buffer falls within the following range [t_p+Δ, t_p+Δ+ω]. [3] Figure 1 depicts a view of packets with relevant timestamps (buffer state) for a peer at an scheduling event. t_p, t_{src}, t_{last} and t_{new} denote peer's and source's playout times, the largest reported timestamp by parents in the last scheduling events, and the largest reported timestamp in this scheduling event.

Careful examination of Figure 1 reveals that the buffer consists of three distinct regions (*i.e.*, range of timestamps) as follows:

- *Playing Region*: this is the left most region with timestamps within the following range [t_p+Δ, t_p+$2 * \Delta$]. We call this playing region since all these packets are being played in the next interval. Therefore, explicitly requesting any missing packet from this region explicitly addresses the timing requirement.
- *New Region*: this is the right most region with timestamps within the following range [t_{last}, t_{new}] which represents all the new packets with largest timestamps that have become available among parents since the last scheduling event. Requesting these packets explicitly expands the pool of new packets which in turn facilitates the diversity of delivered packets to individual peers.
- *Swarming Region*: This is a larger region in the middle. Since there is no preference among missing packets in this region, these missing packets can be requested in a random/rarest-first fashion in order to address diversity. The relatively large size of swarming region provides opportunity to diversify available packets in this region among peers.

Given the above properties of packets in these three regions, the design of a packet scheduling scheme has the following two dimensions: *(i)* the relative priority (*i.e.*, the order of requesting missing packets) of different regions, and *(ii)* the choice of *random* or *rarest-first* strategy to select a subset of packets that are missing but available among parents. Note that setting relative priorities for different regions implicitly controls the allocated packet budget to each region. These two dimensions of design space for packet scheduling scheme motivate the following eight *candidate* scheduling schemes:

- *Rare* or *Rand*: These schemes select all the packets from the entire window using a rarest-first (*e.g.*, Coolstreaming [3] or PULSE [7]) or random (*e.g.*, BitTorrent [2] or Chainsaw [5]) strategy, respectively. By enforcing random/rarest-first strategy across the entire window, these schemes maximize the diversity among delivered packets to different peers. These schemes implicitly address in-time delivery (or timing) of packets since the number of opportunities to request a packet is equal to the number of scheduling events that its timestamp has appeared within the window. This number is larger for packets with earlier timestamp and thus they are more likely to have been requested.
- *PRare* or *PRand*: This scheme explicitly addresses the timing requirement (similar to [4]) by first requesting all

[3] Packets with timestamp [t_p,t_p+Δ] are being played during this interval.

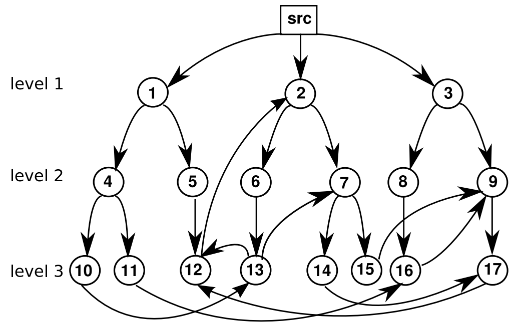


Fig. 2. Organized view of a random mesh

the missing packets in the playing region, and then using the remaining packet budget to select rare/random packets from the rest of the window $[t_p+2*\Delta,\ t_{new}]$

- *NRare* or *NRand*: This scheme explicitly addresses the availability of new packets by first requesting all the new packets (from the new region), and then using the remaining budget to request a rare/random subset of packets from the rest of the window $[t_p+\Delta, t_{last}]$.
- *NPRare* or *NPRand*: This is a hybrid scheme [6] that first requests all the available packets from the new region, then all the missing packets from the playing region, and finally uses any remaining budget to request a rare/random subset of packets from the swarming region. Therefore, this scheme explicitly addresses both timing and availability. [4]

The output of each packet scheduling scheme is an ordered list of required packets that are available among parents and should be requested. The next step is *parent selection* where selected packets are mapped to request from individual parents. Toward this end, we assign each packet to a parent that can provide the packet, and a smaller fraction of its packet budget has been assigned so far. This assignment policy tends to balance the number of assigned packets to individual parents proportional to their packet budgets and exhibits the best performance compared to other policies as we illustrated in our earlier study [6].

Clearly, one can design other scheduling schemes that balances the conflict between timing and diversity differently [8]. However, we believe that our candidate schemes allow us to properly explore the importance of addressing the timing and diversity requirement in an implicit or explicit fashion, and thus identify fundamental design tradeoffs. Furthermore, our candidate schemes adequately resemble most of the commonly used scheduling schemes in previous studies.

IV. Evaluation Methodology

To reliably assess the performance of a live MPS system, we need an evaluation methodology that properly dissects the inherent abilities of a scheduling scheme from the improvement caused by excess resources. Our earlier work on design and evaluation of a new MPS mechanism [6] has inspired the following observation: the evaluation methodology for MPS mechanisms should capture the global pattern of content delivery since this pattern directly determines timing, availability and diversity of delivered packets to participating peers. This in turn provides a useful insight to identify the underlying causes for the observed performance by a packet scheduling scheme.

In this section, first we present a proper view and a set of metrics to capture the global pattern of content delivery. Second, we present the proper pattern of content delivery that maximizes the utilization of resources and derives the key characteristics or *signature* of such a pattern. We then sketch our methodology for evaluating different packet scheduling schemes and present our simulation settings.

Proper View & Performance metrics: We leverage the "organized view" of a randomly connected mesh [6] to properly observe the global pattern of content delivery throughout the overlay. In the organized view, participating peers are grouped into *levels* based on their shortest distance (in hops) from source through the overlay as shown in Figure 2. Peers in level 1 are directly connected to source, peers in level 2 are two hops away from source and so on. Packets of any newly generated segment at sources must be delivered to levels of the overlay in a sequential fashion, *i.e.*, pulled by different peers at level 1, then by different peers at level 2 and so on.

In essence, the organized view clearly illustrates the direction that the packets of each newly generated segment should flow through different levels away from source.

To capture the global pattern of content delivery, we introduce two metrics that are inspired by the organized view of the overlay and are defined on a per-level basis as follows:

- *Diffusion rate* of level i presents the rate by which *new* packets are delivered to (peers in) level i. To capture the diffusion rate of a level, we only capture the first copy of each packet that arrives at that level.
- *Diffusion time* of a packet to level i is the time that elapses from its generation time at source until the first copy of this packet is pulled by a peer in level i. We present the diffusion time in terms of the number of intervals (Δ) to provide an easy comparison with periodic pulling of packets by individual peers.

Since new packets diffuse through the levels of the overlay away from source, any *new* packet that is pulled by a peer in level i must be provided by a parent peer in level $i-1$. Therefore, the diffusion rate of a level represents the rate of availability of new packets to peers in each level whereas the packet diffusion time illustrates how fast the new packets reach that level. In summary, diffusion rate and the distribution of packet diffusion time for individual levels of the overlay collectively capture the global pattern of content delivery through the overlay.

Signature of a Well-behaved Pattern: To provide a reference for examining the performance of different packet scheduling schemes, we present two conditions for the global pattern of content delivery that enables peers to effectively contribute

[4]The other possible hybrid scheduling scheme that gives higher priority to playing region, namely *PNRand* and *PNRare*, has a performance similar to *PRare* and *PRand* , and therefore it is not considered.

their outgoing bandwidth while utilizing their incoming bandwidth from their parents as follows:
1. Required Diffusion Rate, the diffusion rate to *all* levels of the overlay must be equal (or very close) to stream bandwidth. This condition ensures that participating peers in all levels continuously receive new packets. The continuous availability of new packets enables the packet scheduling to achieve higher diversity in available packets among peers [6].
2. Sufficiently Short Diffusion Time (or Large Buffering), the amount of buffering at each peer should be sufficiently large to provide adequate time for in time delivery of required packets. The minimum buffer requirement is equal to the diffusion time of packets to the bottom level (where most of the participating peers and thus most of the system resources are located) in terms of intervals plus the number of intervals for swarming packets of a segment. Earlier studies [3], [6] showed that in a directed and randomly connected overlay, peers require at least three intervals ($3*\Delta$) for swarming packets of a segment. This implies that the minimum buffer requirement is equal to the diffusion time of all (or most) packets to the bottom levels of the overlay, plus three extra intervals. Therefore, any factor that reduces the diffusion time of packets through the levels could directly affect the buffer requirement at each peer. In summary, the diffusion time of packets to the bottom level indicates the required buffering at each peer for a given scheme.

When the above mentioned conditions are met, new packets flow through the overlay at a sufficiently high rate that provides availability. This in turn ensure proper degree of diversity in delivered packets to participating peers to enable effective swarming. Furthermore, the new packets arrive sufficiently early so that there is still plenty of time to swarm packets before their playout times.

Our Methodology: Our evaluation methodology incorporates the following ideas to separate the effect of packet scheduling scheme from available resource on system performance. First, we keep all other components of the system constant and employ the best known practice for those components. More specifically, source incorporates a coordination mechanism and swaps an already delivered packet with a rarest packet within Δ seconds around the requested timestamp. This policy significantly increases the utilization of source bandwidth without over-writing the scheduling scheme by peers We focus on a scenario with homogeneous and symmetric access link bandwidth for participating peers. This implies that all peers should simply receive all packets of the stream. We do not consider churn in our evaluation to avoid any potential side effect that it may have on our findings. In essence, our results represent the best possible performance of the candidate scheduling scheme. We examine all the scheduling schemes over the same randomly connected and directed overlay with the same incoming and outgoing degree for all peers.

Second, we first examine the performance of all the scheduling schemes in a "resource constraint" scenario with minimum source and peer bandwidth, and minimum buffer at each peer. In this scenario, source bandwidth that is minimum value required for the delivery of stream to the top level of the overlay, peer bandwidth is equal to the stream bandwidth, and the available buffer at each peer is equal to the number of levels in the overlay plus three intervals. The resource constrain scenario stress-tests a packet scheduling scheme and exposes its inherent ability to operate without any excess resources in the system. Then, we illustrate how adding different types of excess resources (*i.e.*, source and peer bandwidth) can improve the performance of those schemes the performed poorly with limited resources.

We note that our methodology and our findings are valid for bidirectional overlays and heterogeneous groups as well. We examined these scenarios and their results are available in our related technical report [9] but are not included due to the limited space.

Simulation Setup: To illustrate the proposed evaluation methodology, we investigate the effect of candidate scheduling schemes and available resources using *ns* simulations. Using packet level simulation is a proper choice because it incorporates packet level dynamics, delay and loss. It also enables us to directly control the available resources in the overlay, and thus reliably derive our conclusions. Neither session level simulations nor experiments over PlanetLab does not provide these desired features simultaneously. In our simulations, physical topology is generated by Brite [10] with 15 AS and 10 routers per AS in top-down mode and RED queue management in all routers. Except when noted, we use the following default settings: Δ is 6 seconds, incoming and outgoing peer degree are both 6, overlay is uni-directional, peer population is 200 stream and peer bandwidth are set to 700 Kbps, and all peers have homogeneous and symmetric access link bandwidth. In this scenario, the overlay has 4 levels and the amount of buffering at each peer (ω) is set to its minimum value of 7*Δ. Furthermore, delay on each access link is randomly selected between $[5ms, 25ms]$ while core links have high bandwidth in the range of 4 to 10 Gbps. This ensures that in our simulations bandwidth bottleneck is always at the edge, and avoids any subtle effect of major congestion in the core. Source bandwidth is set to 750 Kbps which is the minimum value for delivery of the stream to peers in level 1. Each simulation is run for 400 seconds.

V. Effect of Scheduling

In this section, we examine the performance of our candidate scheduling schemes in our default "resource-constraint" scenario.

Global Pattern of Content Delivery: Figure 3(a) shows the diffusion rate to the top three levels of the overlay for all the eight candidate scheduling schemes. This figure reveals that all four scheduling schemes that prioritize new packets (*i.e.*, *NP** and *N** schemes) achieve high diffusion rate across all levels of the overlay regardless of other aspects of scheduling scheme. However, other four scheduling schemes are unable to achieve this goal. Figures 3(b) and 3(c) present the distribution of diffusion time across all delivered packets to top two levels (in terms of the number of intervals Δ) and offer a complementary

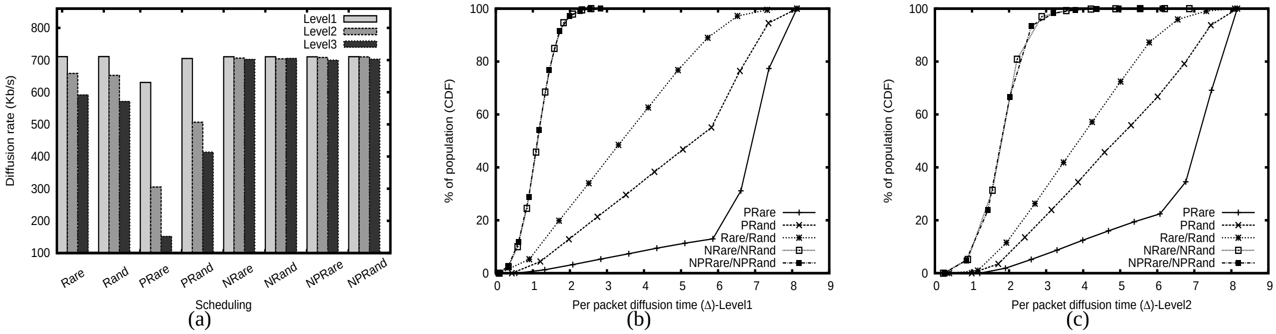


Fig. 3. (a) Diffusion rate across different levels for various scheduling schemes. (b) and (c) Distribution of per-packet diffusion time normalized by Δ for different scheduling schemes, (b) is for level 1 and (c) is for level 2

view of the pattern of content delivery [5]. The diffusion time at these top levels show the following interesting points: First, all the scheduling schemes that prioritize new packets, manage to diffuse the majority of packets to level l within $l+1$ intervals. To explain this, we note that a packet that is generated by source at the beginning of one interval, can be delivered to level 1 by the end of the next interval. Delivery to other levels simply adds an additional interval to the diffusion time as shown in these figures. Figures 3(b) and 3(c) also show that *N** schemes achieve short diffusion time to the bottom level which provides sufficient time for swarming. Therefore, in *N** schemes all peers (regardless of their location in the overlay) can effectively utilize their incoming bandwidth and experience high quality. Figure 4 shows that the distribution of the incoming bandwidth utilization among participating peers is more than 94% for *N** schemes that confirms the above explanation.

Second, the diffusion time for both *Rare* and *Rand* schemes that purely swarm, has a uniform distribution across the entire window (all seven intervals), and does not significantly change across levels. In essence, these figures indicate that in these two schemes, new packets arrive at each level in a totally random order. While all packets arrive at each level within 7 (or ω) intervals, the fraction of packets (*i.e.*, 10%) that arrived at each level during the last interval are late and therefore they are not requested by peers in the next level. This in turn reduces the diffusion rate to lower levels by 10% as shown in Figure 3(a). In summary, late arrival of new packet to the top level has a propagating effect on the diffusion rate of other levels. Moreover, the diffusion time for only half of the delivered packets to the bottom level is sufficiently short to swarm. This results in moderate utilization of incoming bandwidth (around 50%) among participating peers in *Rare* and *Rand* schemes, as shown in Figure 4.

Third, in the two scheduling schemes that explicitly address timing requirement (*P** schemes), roughly 80% (and 50%) of packets experience a very long diffusion time and arrive at the top-level after six intervals. In these two schemes, packets do not properly flow through different levels of the overlay. Closer examination of these schemes reveals that, in *PRare* scheme

[5]The distribution of diffusion time for other levels follow the same trend and are not shown.

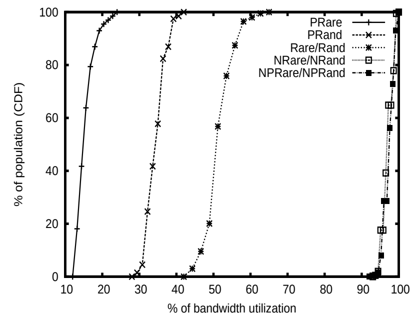


Fig. 4. Distribution of utilization of bandwidth

each peer in the top level can locate most of its required packets in the playing region only at source. These packets are pulled from source around six intervals after their generation time which consumes most of the source bandwidth for late delivery of these packets. The high diffusion time of these packets does not provide sufficient time for peers in lower levels to pull them. This reduces the diffusion rate to lower levels, limits the availability of new packets among those peers and limits their ability to effectively swarm which leads to poor utilization of their incoming access link bandwidth as shown in Figure 4. The random nature of selection in *PRand* scheme somewhat addresses this problem and slightly improves the flow of new packets through the overlay. In summary, the main drawback of scheduling schemes that explicitly address timing requirement (*i.e.*, *P** schemes) is their inability to properly utilize the available bandwidth from parents in higher levels to diffuse new packets in order to ensure availability and thus diversity of new packets among peers.

Content Availability for Individual Peers: We now examine the availability of content to individual peers in order to better understand the dynamic of content delivery for different scheduling schemes. Figure 5(a) shows the average percentage of available packets (including those that have been received) among parents of individual peers for all candidate scheduling schemes. We further divide the availability of packets to different sub-windows of length Δ to show the variations over different range of timestamps. This figure basically represents the "average view" of available packets to each peer across different range of timestamps. The figure reveals a wide gap in the availability of content to a peer in different scheduling schemes, especially in sub-windows with lower timestamp.

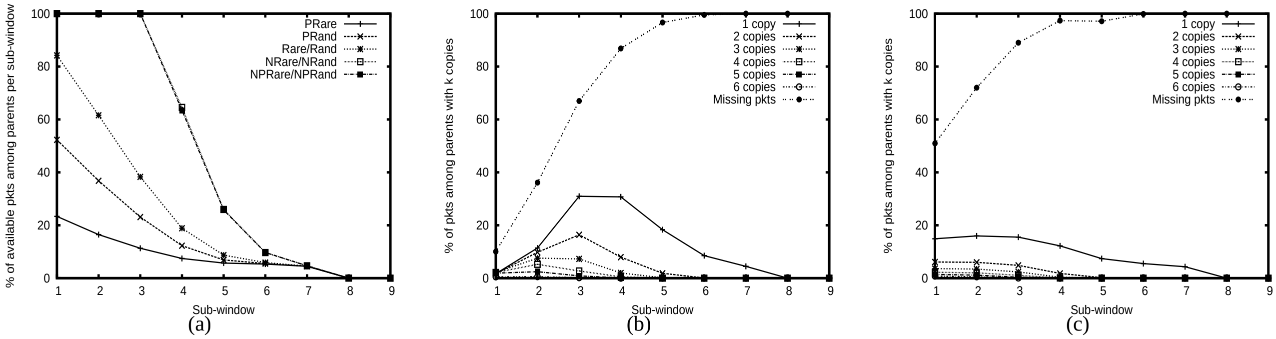


Fig. 5. (a) Percentage of available packets among parents across different sub-windows for various scheduling schemes. (b) and (c) Average percentage of number of copies of each available packet among parents that is required-but-missing for $N*$ and $Rare/Rand$ schemes, respectively.

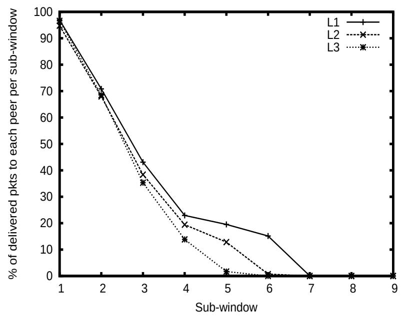


Fig. 6. Distribution of average delivered packets in each peer for $N*$ schemes

Those scheduling schemes that prioritize new packets (*i.e.*, $N*$ schemes) achieve a significantly higher degree of content availability than others. Pure swarming schemes still outperform the schemes that prioritize playing packets. This figure clearly confirms our earlier observation and indicates that poorly performing schemes experience "content bottleneck" among parents.

Figure 5(b) and 5(c) present the diversity of required-but-missing packets among parents of each peer for $N*$ and *Rare* /*Rand* schemes, respectively. These figures plot the average percentage of packets with k copies in each sub-window (across different scheduling events) for different values of k. We also show the average percentage of missing packets by a peer within each sub-window which roughly presents the probability of requesting a new packet from a given sub-window (*i.e.*, average demand for packets within each sub-window). These two figures illustrate two important points: First, in both $N*$ and *Rare* /*Rand* schemes, a significant portion of available packets are unique (*i.e.*, have a single copy). Therefore, random packet selection is likely to select unique packets. This explains the similarity in the performance of schemes that selects packets by rarest-first or random strategy *i.e.*, *Rare* and *Rand* schemes (also *NRare* and *NRand*). Second, comparison between these two figures also demonstrates that prioritizing the new packets not only increases the overall availability of packets but also results in a higher diversity of available packets among peers since most of the available packets for each peer are unique, *i.e.*, availability of wider range of packets leads to a higher degree of content diversity among peers.

Local Pattern of Delivery: Another interesting question is "the pattern that required packets in a window arrive at each peer?". Figure 6 shows the average percentage of delivered packets in each sub-window of the buffer among all peers in a particular level of the overlay in $N*$ scheduling schemes. Since the window slides by Δ seconds (a sub-window) once every Δ seconds, the difference between sub-windows i and $i + 1$ demonstrates the progress in download during sub-window $i + 1$. Figure 6 shows that the rate of progress for different levels are slightly different. During the first $depth$ (three in this example) intervals (or sub-windows), peers in each level sequentially receive a fraction (namely $\frac{1}{deg}$% or 16% in this example) of the packets within their last sub-window. This corresponds to the diffusion rate to each level. During the last ω-$depth$ sub-windows (three in this example), all peers experience a rapid rate of progress and receive an equal fraction of remaining packets in each sub-window. During these sub-windows some packets are available at each peer and swarming occurs.

The above findins collectively illustrate that while the $N*$ schemes achieve high diffusion rate through all levels, most of the content delivery actually occurs during the swarming intervals.

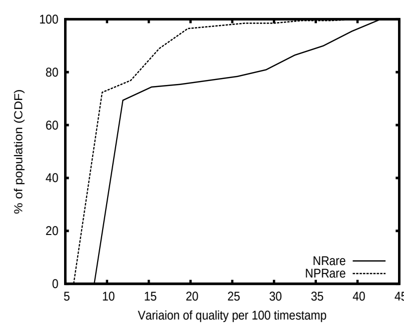


Fig. 7. Distribution of missing packets for $N*$ schemes

Importance of Explicit Timing: Our results indicated that all *N** scheduling schemes that prioritize new packets similarly exhibit good performance. This raises the question "*whether explicitly requesting the required packets in the playing window has any effect on the performance of $N*$ schemes?*", *i.e.*, is there any difference between *N** and *NP** schemes. Figure 7 depicts the distribution in the percentage of missing (or undelivered) packets that represent the stability of delivered quality among all peers in *NPRand* and *NRand* schemes. This figure shows that the percentage of missing packets for those schemes that explicitly request playing packets ($NP*$) is around 10% higher than those implicitly address timing requirements. This difference is due to the fact that while there are multiple opportunities for requesting each packet in *N** schemes, there is still a small chance that some of the required packets are not requested. The schemes that explicitly request the missing packets in the playing window can fill these holes and ensure the stability of delivered quality.

VI. Effect of Resources

We now turn our attention to the effect of excess resources, namely source bandwidth and peer bandwidth, on the performance of live mesh-based P2P streaming. We focus on two scheduling schemes that performed rather poorly in the resource constraint scenario, namely *Rand* and *Rare* , and examine how the availability of more resources (*i.e.*, source and peer bandwidth) affects their performance. More specifically, we investigate both the independent and combined effect of excess source and peer bandwidth on system performance by quantifying their impact on diffusion rate and diffusion time across different levels of the overlay. We show the results for *Rare* , and the results for *Rand* are very similar. We use minimum buffering in these scenarios to eliminate any side effect of buffer on our analysis. The effect of buffer size is later examined in this section. To minimize the effect of overlay connectivity on our results, we keep the peer connectivity constant across these scenarios. This implies that increasing source bandwidth proportionally increases source degree (*i.e.*, number of peers that directly connect to source) whereas increasing peer bandwidth has an opposite effect and proportionally decreases source degree [6].

Effect of Source Bandwidth: Figures 8(a) (only lines that labeled Level i) and 8(b) depict the changes in the diffusion rate and 90-percentile packet diffusion time for top three levels, respectively, when we increase source bandwidth in the resource constraint scenario. The x-axis shows the percentage of increase in source bandwidth. Figure 8(a) clearly illustrates that the diffusion rate to all three levels increases with source bandwidth until they are saturated. Figure 8(b) reveals the underlying reason for the increase in diffusion rate. Increasing source bandwidth directly decreases the diffusion time of packets to level 1. This in turn reduces the fraction of late packets and allows peers in level 2 to pull more packets from peers in level 1. The drop in diffusion time to level 1 also has a ripple effect on the diffusion time of packets to lower levels and similarly increases the diffusion rate of those levels as well.

Note that the aggregate bandwidth for the delivery of new packets to level 1 is limited by source bandwidth which is the main determining factor on the diffusion rate for level 1. However, the aggregate bandwidth between consecutive levels is proportional with the number of connections between them (or roughly the number of peers in the lower level) which is very large. For example, in a scenario with peer degree 6, the total number of connections to level 1, 2 and 3 are 6, 36 and 216 connections, respectively. Therefore, the main performance bottleneck for diffusion rate to other levels (except level 1) is the amount of content and its diffusion time at the higher level (as opposed to the available bandwidth from the higher level). This observation explains the faster increase in the diffusion rate of lower levels with source bandwidth. More specifically, as diffusion time of packets to a level decreases, the abundant available bandwidth to the next level can be utilized more effectively, and causes an even bigger reduction on the diffusion time of the next level.

Effect of Peer Bandwidth: Increasing peer bandwidth (since peer degree and source bandwidth remain constant) simply increases the available bandwidth of individual connections (per-connection bandwidth) in the overlay. This proportionally increases the aggregate bandwidth between levels but does not have any effect on the bandwidth from source to level 1. To examine the effect of peer bandwidth on performance, we double peer bandwidth in the resource constraint scenario, and call this scenario high peer bandwidth (labeled as HPBW) scenario. The six data points on the y-axis of Figure 8(a) present the diffusion rate of the three levels for both the resource constraint and HPBW scenarios. Our results clearly illustrate that doubling peer bandwidth has a negligible effect on the diffusion rate or even diffusion time. This may seem surprising because increasing peer bandwidth for all participating peers significantly increases the aggregate resources in the system compared to increasing source bandwidth alone. However, this result supports our earlier explanation. Since the available bandwidth between levels is already abundant, and increasing peer bandwidth does not change the diffusion rate or diffusion time to level 1 that are the main performance bottleneck and are not affected despite this dramatic increase in available resources.

Combined Effect of Source and Peer Bandwidth: Figure 8(a) and 8(c) demonstrate the combined impact of source and peer bandwidth by showing the diffusion rate and diffusion time as a function of source bandwidth when peer bandwidth is doubled (*i.e.*, HPBW scenario). Figure 8(a) clearly shows that increasing peer bandwidth does not have any effect on the diffusion rate. This again confirms our observation that even in the resource constraint case, the aggregate available bandwidth between levels is sufficient. Further increase in the per-connection bandwidth does not lead to any further improvement in the diffusion rate.

Comparing Figure 8(b) and 8(c) reveals that as source bandwidth increases, higher peer bandwidth can further drop the diffusion time to lower levels. To explain this, we recall that increasing source bandwidth rapidly drops the diffusion time to level 1 (as we showed in Figure 8(b)). As source bandwidth increases, the delivered packets to level 1 experience shorter

[6] We note that any change in the degree of source directly affect the depth of the overlay and changes the required buffering at each peer. However, since our methodology focuses on the pattern of delivery through levels, these changes can be captured by our methodology and does not affect our discussion

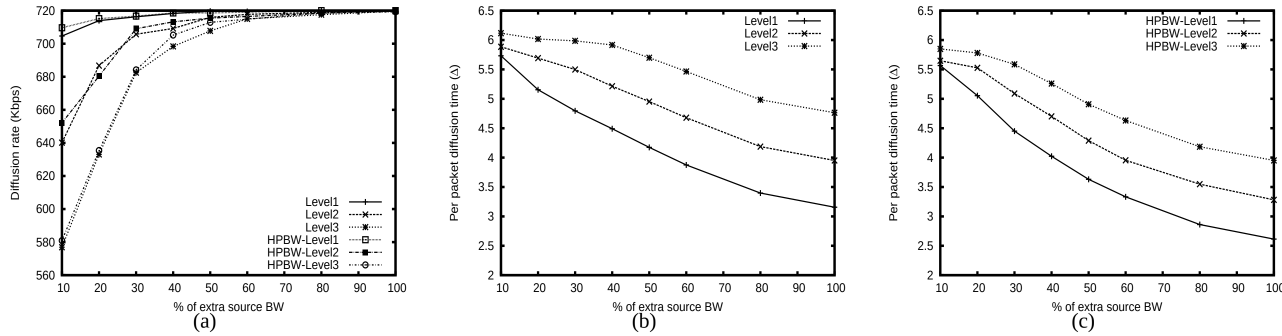


Fig. 8. (a) Diffusion rate of different levels as source bandwidth increases. (b) and (c) 90-percentile of per-packet diffusion time of different levels as source bandwidth increases with minimum peer bandwidth in figure (b) and doubled peer bandwidth in figure (c)

diffusion time but the available content to level 1 is always the bottleneck. However, the faster availability of new packets to level 1 provides more time for delivery of packets to other levels, and thus enables the system to deliver the same number of packets within a shorter period of time without changing the diffusion rate to lower levels. This in turn enables the system to operate with a smaller amount of buffering.

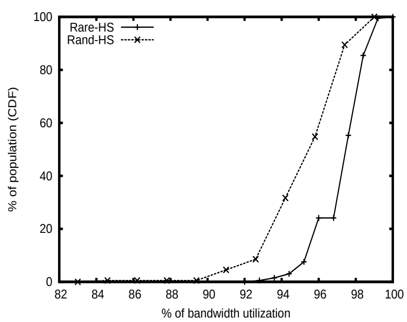


Fig. 9. of Distribution of bandwidth utilization for *Rare* and *Rand* schemes

Taking a Closer Look: Our evaluation methodology effectively captures the global pattern of content delivery which represent the primary factors that affect the overall performance. However, there might be some minor differences between two scheduling schemes that are not detected by our metrics. Figure VI depicts the distribution bandwidth utilization for *Rare* and *Rand* schemes where source bandwidth is doubled. While all other characteristics of these two schemes were pretty similar, this figure indicates that peers experience a slightly better performance with *Rand* scheme when this excess resources become available. This illustrates the complexity of performance evaluation in mesh-based P2P streaming.

In summary, our results in this section collectively illustrate that increasing source bandwidth has a significant effect on system performance because it directly increases diffusion rate to level 1. This also reduces the diffusion time to level 1 and has a ripple effect on the diffusion rate of lower levels. In contrast, increasing peer bandwidth has a rather limited effect since the performance is limited by the the diffusion rate and available content at level 1 that are determined by source bandwidth, and are not affected by peer bandwidth at all. Since increasing source bandwidth only affects a single point which is often controlled by the content provider (*i.e.*, the source) and has a much larger impact for the unit of excess bandwidth, it seems to be an obvious choice to improve the performance of mesh-based P2P streaming systems. Furthermore, our findings clearly demonstrate how by adding the right amount of proper excess resources, one could get any poorly-designed scheduling scheme to exhibit a good performance.

VII. Extended Evaluations

A. Peer degree

In this section, we examine the performance of our candidate scheduling schemes in our default "resource-constraint" scenario with peer degree of 4 and 10.

Figures 10(a) shows the diffusion rate to the top three levels of the overlay for all the eight candidate scheduling schemes when peer degree is 4. Figures 10(b) and 10(c) present the distribution of diffusion time across all delivered packets to top two levels (in terms of the number of intervals Δ) for peer degree 4. Sub-figures in 11 show the same results for peer degree 10. Overall these reveal that peer degree does not have a major impact on the diffusion rate and time in *NP**, *N** and *Rare* or *Rand* schemes. However, peer degree affects the diffusion time and rate of packets in *PRare* and *PRand* schemes. Increasing the number of level 1 peers magnifies the role of source coordination. Each level 1 peer has a fixed packet-budget to request from source, increasing number of peers in level 1 (source children) reduces packet-budget to each one of them while keeps the aggregate budget fixed. Therefore, the probability of requesting redundant packets specially from playing sub-window increases which results in decrease of diffusion rate to level 1.

The rest of the results show similar trend which approve our previous findings for degree 6 and reveals that regardless of peer degree our findings are valid.

B. Bi-directional overlay

In this section, we examine the performance of our candidate scheduling schemes in our default 'resource-constraint" scenario with bi-directional overlay.

Figure 13(a) shows the diffusion rate to the top three levels of the overlay for all the eight candidate scheduling schemes when overlay is bi-directional. Figures 13(b) and 13(c) present the distribution of diffusion time across all delivered packets

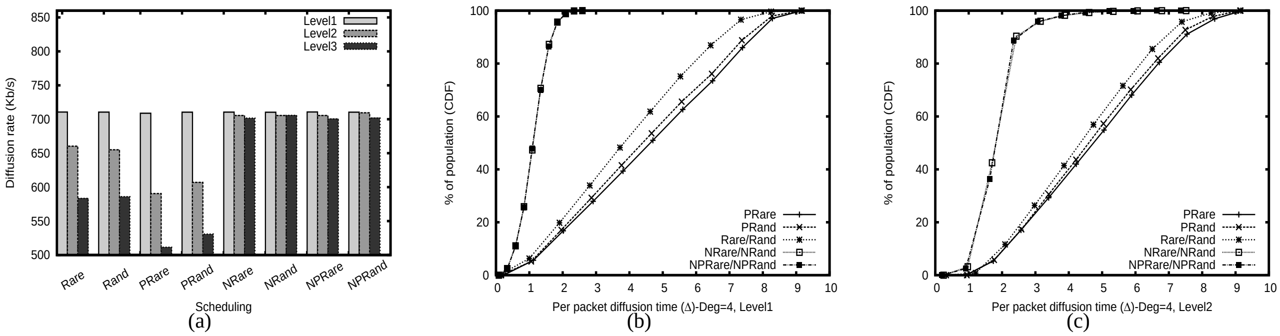


Fig. 10. (a) shows diffusion rate across different levels for various scheduling schemes with peer degree 4. (b) and (c) show the distribution of per packet diffusion time for various scheduling schemes when peer degree is 4 across level 1 and 2 respectively.

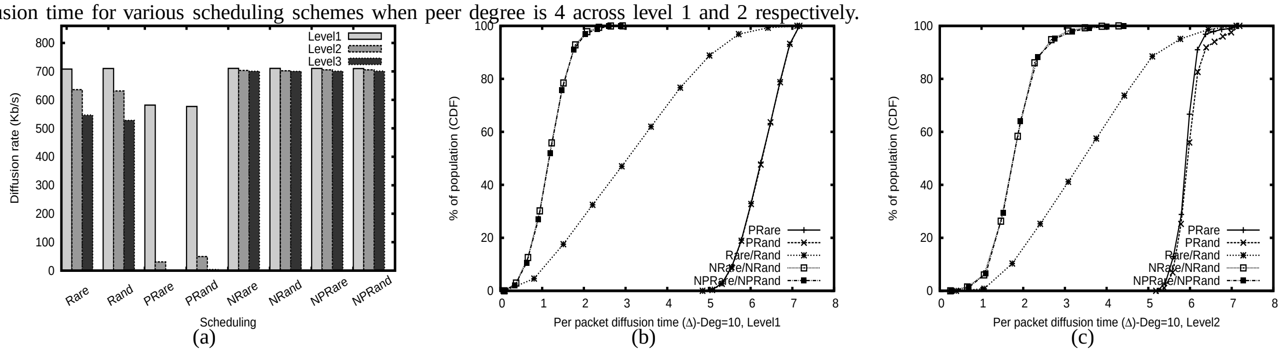


Fig. 11. (a) shows diffusion rate across different levels for various scheduling schemes with peer degree 10. (b) and (c) show the distribution of per packet diffusion time for various scheduling schemes when peer degree is 10 across level 1 and 2 respectively.

to top two levels (in terms of the number of intervals Δ) for a bi-directional overlay. We can observe that bi-directionl overlay does not have any impact on diffusion time and rate to different levels of the overlay for various scheduling schemes.

Figure 14(a) shows the distribution of bandwidth utilization for various scheduling schemes in a bi-directional overlay. This figure reveals the same trend in the performance of various schemes. Moreover, by comparing figures 4 and 14(a) we can observe that in a bi-directional overlay regardless of scheduling schemes utilization of bandwidth decreases. Figure 14(b) presents the percentage of available packets among parents across different sub-windows for various scheduling schemes in a bi-directional overlay. This figure also confirms our previous findings about diversity of packets among parents for different scheduling schemes. Clearly, a comparison between this figure and Figure 5(a) reveals that in a bi-directional overlay diversity (percentage of available packets among parents) regardless of scheduling schemes decreases. The rest of the results show similar trend which approve our previous findings for uni-directional overlays and reveals our findings are still valid in bi-directional overlays.

C. Distorted overlay

In this section, we examine the performance of our candidate scheduling schemes in our default ‘resource-constraint” scenario with distorted overlays. After the system reaches its steady state and peers fully connect to each other, we randomly remove X% of peers from the overlay simultaneously without further repairing it. The performance of the content delivery in such distorted overlays represents the performance in the worst case scenario in presence of churn/peer-departure.

Figure 15(a) shows the diffusion rate to various levels for different scheduling schemes after the departure of 10% of peers from the system. Figures 15(b) and 16(b) reveal the distribution of diffusion time across all delivered packets to top two levels (in terms of the number of intervals Δ) for 10% of distortion. Figures 16 and 17 show the same set of results for 20% and 30% distortion in the overlay. From these figures we can observe that the behavior of various scheduling schemes does not change by different levels of distortion in the overlay.

Figure 18(a) shows the median, 5th and 90th percentile of percentage of bandwidth utilization for various scheduling schemes across different levels of distortion. Clearly by increasing the distortion bandwidth utilization decreases. Figure 18(a) reveals that all of the scheduling schemes show the same trend. The rest of the results show similar trend which confirm our previous findings for fully connected overlays.

VIII. Conclusion and Future Work

This paper examined the role of packet scheduling and available resources on the performance MPS mechanism for live content. Our key observation is that the global pattern of content delivery determines the performance of a MPS mechanism. Using this observation, we presented a new methodology to capture the global pattern of delivery and evaluate the performance of MPS mechanism. We illustrated the ability of our methodology by assessing the performance of eight packet scheduling schemes that represent a wide range of MPS mechanisms. Our methodology provides a unified framework

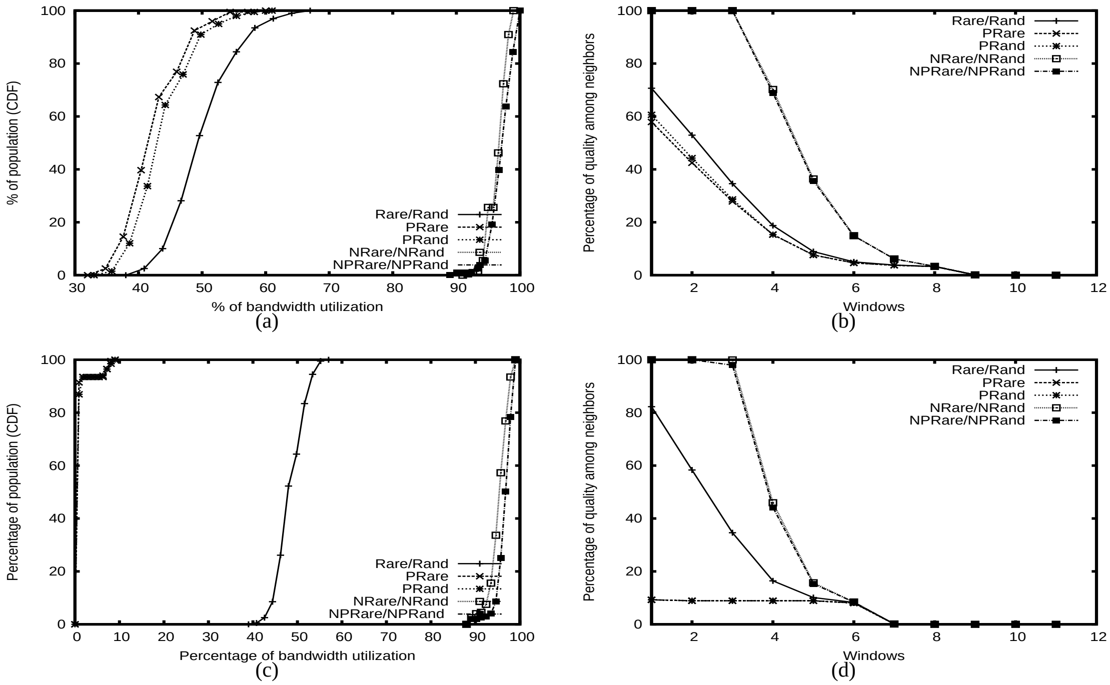


Fig. 12. (a) and (c) distribution of utilization of bandwidth across various scheduling schemes for degree 4 and 10, respectively. (b) and (d) Percentage of available packets among parents across different sub-windows for various scheduling schemes for degree 4 and 10, respectively.

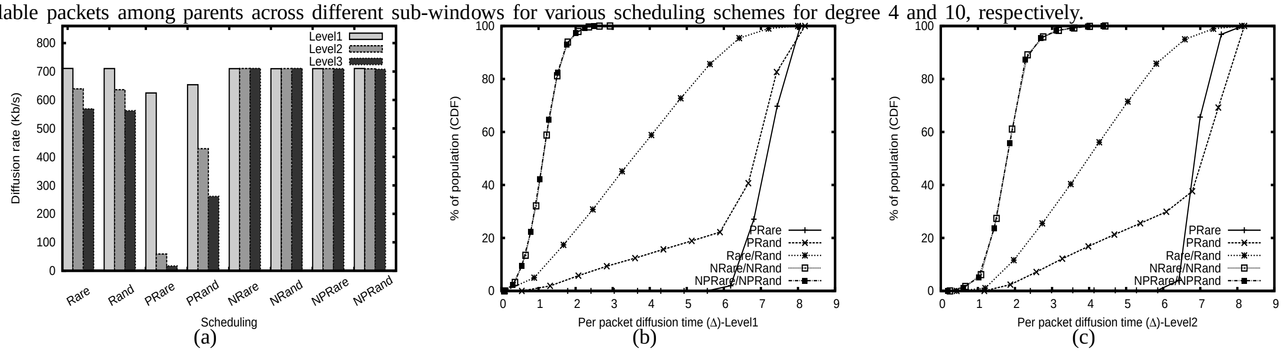


Fig. 13. (a) shows diffusion rate in a bi-directional overlay across different levels for various scheduling schemes. (b) and (c) show the distribution of per packet diffusion time in a bi-directional overlay for various scheduling schemes across level 1 and 2 respectively.

for comparing different MPS mechanism and she an insightful light on their design and evaluations.

REFERENCES

[1] N. Magharei, R. Rejaie, and Y. Guo, “Mesh or Multiple-Tree: A Comparative Study of P2P Live Streaming Services,” in *INFOCOM*, 2007.

[2] B. Cohen, “Bittorrent.” [Online]. Available: http://www.bittorrent.com

[3] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, “Coolstreaming: A data-driven overlay network for live media streaming,” in *INFOCOM*, 2005.

[4] M. Zhang, Y. Xiong, Q. Zhang, and S. Yang, “On the optimal scheduling for media streaming in data-driven overlay networks,” in *GLOBECOM*, 2006.

[5] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, “Chainsaw: Eliminating Trees from Overlay Multicast,” in *IPTPS*, 2005.

[6] N. Magharei and R. Rejaie, “PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming,” in *INFOCOM*, 2007.

[7] F. Pianese, J. Keller, and E. W. Biersack, “Pulse, a flexible p2p live streaming system,” in *GI*, 2006.

[8] A. Vlavianos, M. Iliofotou, and M. Faloutsos, “Bitos; enhancing bittorrent for supporting streaming applications,” in *GI*, 2006.

[9] N. Magharei and R. Rejaie, “Dissecting the performance of Live Mesh-based P2P Streaming,” Tech. Rep. CIS-TR-07-05, 2007. [Online]. Available: http://mirage.cs.uoregon.edu/pub/tr07-05.pdf

[10] A. Medina, A. Lakhina, I. Matta, and J. Byers, “BRITE: An Approach to Universal Topology Generation,” in *MASCOTS*, 2001.

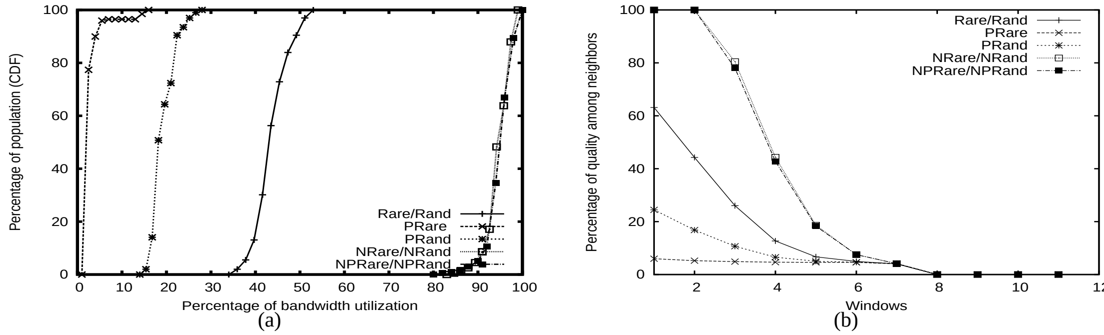


Fig. 14. (a) shows the distribution of utilization of bandwidth across various scheduling schemes for bi-directional overlay. (b) is the percentage of available packets among parents across different sub-windows for various scheduling schemes for bi-directional overlay.

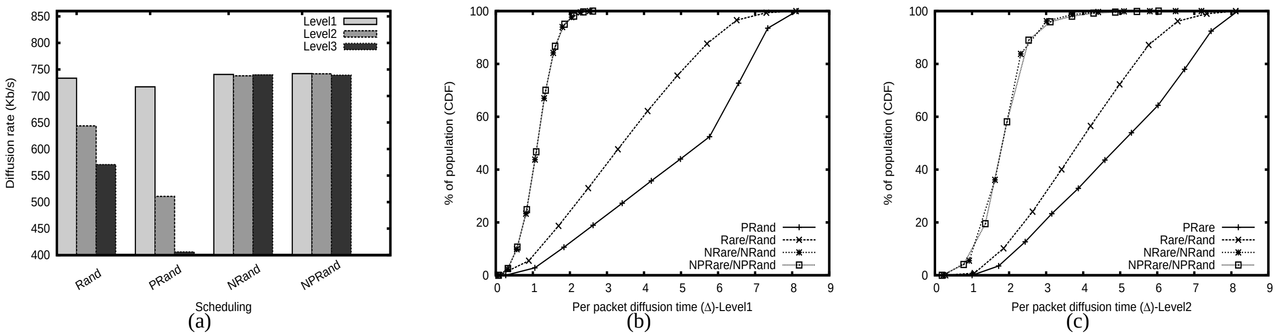


Fig. 15. (a) shows diffusion rate across different levels for various scheduling schemes when 10% of peers leave the overlay. (b) and (c) show the distribution of per packet diffusion time for various scheduling schemes when 10% of peers leave the overlay across level 1 and 2 respectively.

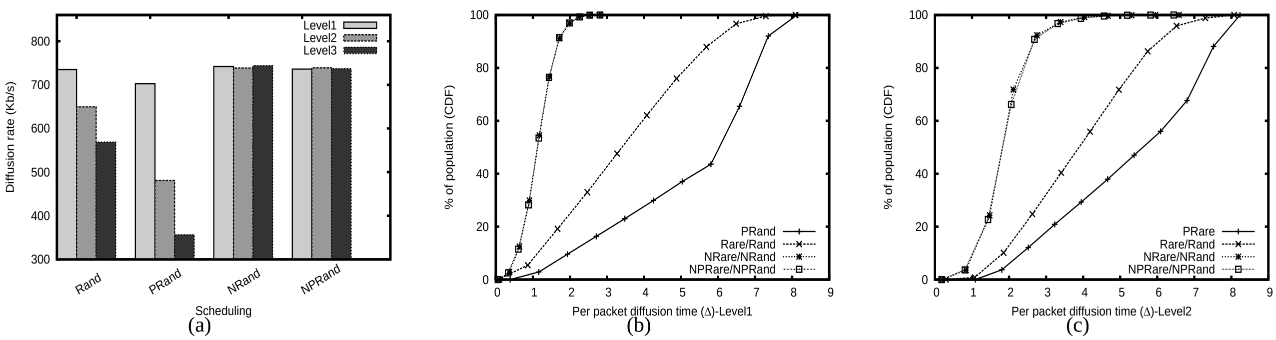


Fig. 16. (a) shows diffusion rate across different levels for various scheduling schemes when 20% of peers leave the overlay. (b) and (c) show the distribution of per packet diffusion time for various scheduling schemes when 20% of peers leave the overlay across level 1 and 2 respectively.

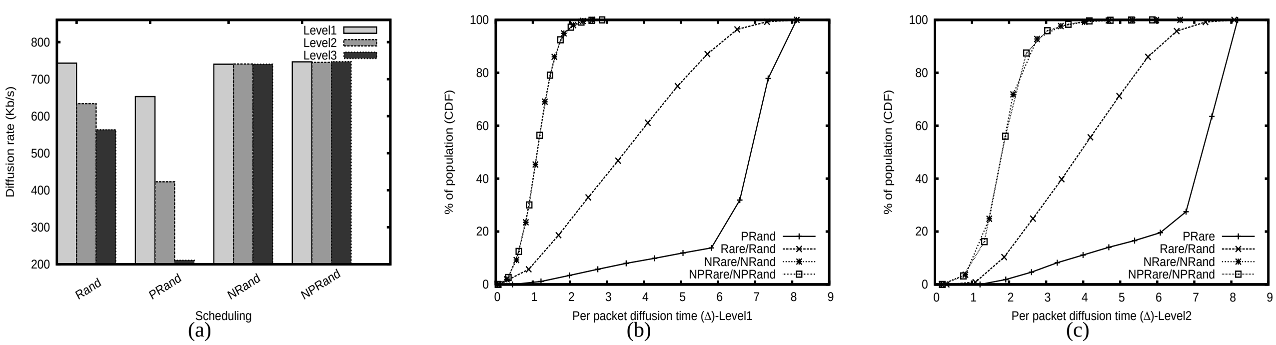
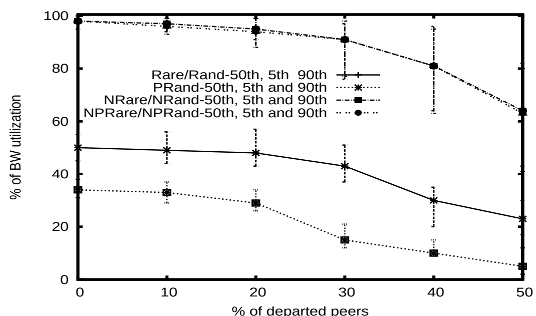


Fig. 17. (a) shows diffusion rate across different levels for various scheduling schemes when 30% of peers leave the overlay. (b) and (c) show the distribution of per packet diffusion time for various scheduling schemes when 30% of peers leave the overlay across level 1 and 2 respectively.



(a)

Fig. 18. (a) shows the percentage of utilization of bandwidth across various scheduling schemes for different percentage of departed peers.