Push-Pull Incentive-based P2P Live Media Streaming System

POO KUAN HOONG, HIROSHI MATSUO
Nagoya Institute of Technology
Department of Computer Science and Engineering
Showa-ku, Gokiso, Nagoya, 466-8555
JAPAN
khpoo@matlab.nitech.ac.jp, matsuo@nitech.ac.jp

Abstract: Peer-to-peer (P2P) file sharing has become increasingly popular, accounting for as much as 70% of Internet traffic by some estimates. Recently, we have been witnessing the emergence of a new class of popular P2P applications, namely, P2P audio and video streaming. While traditional P2P file distribution applications target elastic data transfers, P2P streaming focuses on the efficient delivery of audio and video content under tight timing requirements. In these applications, each node independently selects some other nodes as its neighbors and exchanges streaming data with neighbors. In this paper, we propose and investigate a full distributed, scalable, and cooperative protocol for live video streaming in an overlay peer-to-peer network. Our protocol, termed *P2P Unstructured Live Media Streaming (PALMS)*, makes use of combination of push-pull score-based incentive method to achieve high performance (in term of delay, stream continuity, cooperation, etc.). The main contribution of PALMS is that it reduces the end-to-end streaming delay and in turn results better delivered quality. Furthermore, with the implementation of score-based incentive mechanism, PALMS is resilient to existence of free-riders and encourage cooperation among participating nodes. We have extensively evaluated the performance of PALMS. Our experiments demonstrate that PALMS achieves good streaming quality even under the existence of free-riders.

Key–Words: peer-to-peer, streaming, free-riders, overlay, push-pull

1 Introduction

Peer-to-peer (P2P) file sharing has become increasingly popular, accounting for as much as 70% of Internet traffic by some estimates. Recently, we have been witnessing the emergence of a new class of popular P2P applications, namely, P2P audio and video streaming. While traditional P2P file distribution applications target elastic data transfers, P2P streaming focuses on the efficient delivery of audio and video content under tight timing requirements. Still in its infancy, both live and on-demand P2P streaming have the potential of changing the way we watch TV, providing ubiquitous access to a vast number of channels, personalizing your TV experience, and enabling roaming TV services. For a long time, traditional approaches that are client/server based e.g., Akamai [3] have been used for streaming multimedia applications over the Internet.

Over the past few years, P2P networks have emerged as a promising approach for distribution of multimedia content over a network. Some P2P network related research is by the following authors [11],[13],[15],[20],[22]. One form of P2P network, the peer-to-peer overlay, offer a promising approach to support one-to-many multimedia streaming applications without any special support from the network, called P2P streaming. The basic building blocks for P2P streaming, called *nodes* or *peers*, are no longer passive receivers of data but can act both as clients and servers at the same time. Stream data are simultaneously received, played, and passed to other connected peers. The goal of P2P streaming mechanisms is to maximize delivered quality to individual peers in a scalable fashion despite the heterogeneity and asymmetry of their access link bandwidth. An effective P2P streaming mechanism depends on the effective utilization of the outgoing bandwidth of most participating peers.

In this paper, we propose and study a fully distributed, scalable and cooperative protocol for P2P streaming live and non-interactive media streaming, called PALMS (*P2P Unstructured Live Media Streaming*). Similar to DONet [22], PALMS is based on data-driven and receiver-based unstructured overlay media streaming. PALMS is designed to operate in conditions where nodes have heterogenous bandwidths and resources. In comparison to DONet, which only employs pure pull method, PALMS employs a combination of two methods for media streaming, namely the pull method and push method.

The rest of the paper is organized as follows: In

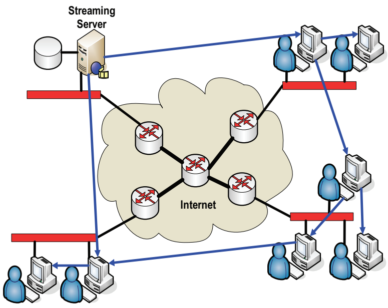


Figure 1: The organized view of PALMS that consists of 7 nodes

Section 2, we discuss the overview system of PALMS. In section 3, we discuss the analysis of pure pull streaming method and detailed algorithms of push and pull packets scheduling. We describe the details of the simulation setting and performance metrics in Section 4. Section 5 provides the result of performance evaluation based on simulation in various conditions. Finally we present our conclusions and discussion on future works in Section 6.

2 PALMS : System Overview

PALMS is based on data-driven and receiver-based unstructured overlay media streaming. It is designed to operate in scenarios where the nodes have heterogenous and variable bandwidth resources. For the ease of exposition, we refer to the media source as the *streaming server* and receivers as *clients*. The term *peers* and *nodes* are interchangeable, and refer to the all the clients. Peers in PALMS implement data exchange policies that enforce and reward node cooperation through a score-based incentive mechanism. The incentive mechanism encourages cooperation among participating nodes and allows gradual improvement in the streaming of media content and in turn improves the delivered quality of streaming.

PALMS consists of three major components: (i) **overlay construction mechanism**, which organizes participating peers into an overlay; (ii) **streaming scheduling mechanism**, which determines the delivery of content from the streaming source to individual nodes through the overlay; and (iii) **incentive mechanism**, which determines service received by a peer as a function of the bandwidth contributed by the peer. In the following subsections, we describe the components in PALMS.

2.1 Overlay Construction

In PALMS, nodes are functionally identical. They are free to exchange control information and media content data from the stream. Participating nodes form a randomly connected directed graph overlay network as shown in Fig. 1. Each peer maintains a certain number of connected nodes that are known as *neighbors*. Each node receives media content from a certain number of parent nodes and relays the content to a certain number of child nodes. *Incoming* and *outgoing* degree denotes the number of connected parents and child nodes respectively.

The basic task of the overlay construction mechanism component for each node is to be in charge of finding appropriate neighbors for each node through the gossip method so that the application layer network can be successfully built up. To join the streaming session, a new peer contacts the bootstrapping node, (streaming server in the case of PALMS) to learn about other participating peers upon arrival. This could be regarded as the login process. The bootstrapping node returns a list of randomly selected peers that can potentially serve as parent nodes. The new peer contacts these potential parent nodes to determine whether they are able to accommodate a new child node. This is by determining whether the parent node still has enough allocation slots on the outgoing degree. The peer also maintains a target incoming degree. If a new peer cannot identify a sufficient number of parents from the reported list, it will contact the bootstrapping node again to obtain another list of potential parent nodes. A successful new peer is registered with the bootstrapping node and can be selected as the parent node by other peers. Each new joined node synchronizes the local clock with the bootstrapping node during login process.

All the nodes will self-organize into an unstructured mesh. Each node has a member table that contains a list of neighbor nodes obtained from bootstrapping node. The information in member tables is encapsulated into a UDP packet and exchanged among neighbors periodically. Each node updates its member table in accordance with the member table sent by its neighbors. Each node sends a periodical *heartbeat* message to update its neighbors. If a node does not update its neighbors periodically, it will be removed from the member table. Once a node leaves, it will broadcast a “leave message” to all its neighbors. The nodes that receive this message will delete the respective node from its member table as well. Therefore, the failure of any neighbors can be detected by constantly monitoring periodical messages from neighbors.

In order to locate a better neighbor, which has

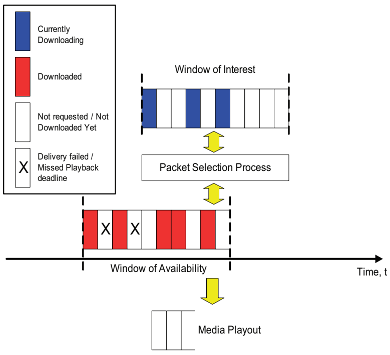


Figure 2: Data buffer for PALMS node

higher uplink, a peer in PALMS periodically replaces the neighbor with the least contribution by selecting nodes with higher scores. This operation helps each node maintain a stable number of partners in the presence of node departures, and it also helps to discourage the existence of free riders within the network.

2.2 Streaming Scheduling

PALMS employs a swarm-like content delivery mechanism that is similar to BitTorrent [8]. The main advantages of swarming content delivery are its ability to effectively utilize the outgoing bandwidth of participating peers and its robustness against the dynamics of peers arrival and departure, which is also known as *churn*.

The streaming scheduling mechanism of each node is responsible for exchanging packets with all its neighbors. Swarm-like content delivery is incorporated in PALMS. As a parent, each peer periodically generates a report i.e., *buffer map* of its newly received packets and sends it to its child nodes. As a child, each peer periodically requests a subset of required packets from each parent based on the reports received. The pull mode is deployed to fetch absent packets from its parent nodes and in turn tries its best to deliver packets requested by the neighbors. Packets requested by the pull mode are determined by the packet scheduling algorithm, which is much similar to the data-driven approach in DONet [22]. Peer selection for PALMS depends on the rank ordering of the score-based incentive mechanism.

Every node also maintains a *window of interest*, which is the set of sequence packets that the node is interested in acquiring at the current time. Figure 2 illustrates the fundamental concept of the sliding window. A sliding *window of availability* contains the list of segments available for each node. This is the information for the *buffer map* shared with other neighbor nodes. The node slides its window of interest forward over time as new packets stream in. If a packet has not

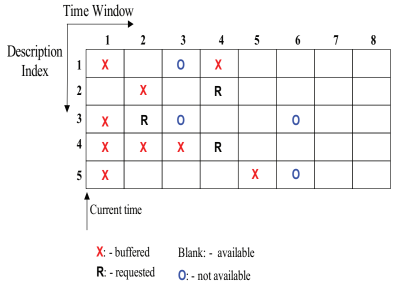


Figure 3: Buffer State at a given time

been received by the time it "falls off" the trailing edge of the window, the node will consider that packet lost or obsolete and will no longer try to acquire it. Figure 3 shows the buffer state of a node at a given time.

To accommodate the bandwidth heterogeneity among peers, the content is encoded with Multiple Description Coding (MDC). Basically, MDC organizes the streaming content into several sub-streams where each sub-stream can independently decoded. The use of MDC for video streaming has been widely studied. Padmanabhan *et al.* propose that introducing redundancy can provide robustness in media streaming [15]. The delivered quality to each peer is proportional to the number of independent sub-streams it receives. With MDC coding, each peer is able to receive the proper number of sub-streams that are delivered through the combination push-pull streaming mechanism.

2.3 Incentive mechanism

PALMS, just like any other P2P content delivery system, works on the premise that peers share resources in order to increase the total capacity of the P2P system. In the case of PALMS, it derives bandwidth from participating peers who operate independently of each other. A mechanism that creates favorable incentives for all peers to contribute resources and thus guards against bandwidth starvation in PALMS is needed to sustain peer interest in sharing bandwidth.

We believe that peer selection for receiving packets at child nodes offers an unique opportunity to tackle both the free-riders and the streaming Quality of Service (QoS) challenges in a synergistic manner. We propose a score-based incentive mechanism that provides service differentiation in peer selection for P2P streaming. Our proposed incentive mechanism is an extension work of [12]. Contributors to the system are rewarded with flexibility and choice in peer selection. Free-riders are given limited options in peer selection, if any, and hence result low quality streaming.

We consider that PALMS consists of rational users who choose their contribution level in order to maximize their individual utility. The contribution level x_i of user i is converted to score S_i, which in turn mapped into a percentile rank R_i. The scoring function used in PALMS is based on the ratio of amount bytes uploaded over bytes download. Peer selection depends on the rank ordering of the requestors and candidate suppliers. For example, a peer selection scheme may allow a user to select peers with equal or lower rank to serve as suppliers. The outcome of the peer selection process is the realized quality of the streaming session. User utility U_i is a function of the streaming session quality Q and the contribution cost C:

$$U_i(x_i) = \alpha_i Q(x_i) - \beta_i C(x_i), \tag{1}$$

where α_i and β_i define the values of streaming quality and contribution cost to user i.

To evaluate delivery quality and quantify the performance of the media streaming system, we define quality, Q of a streaming session as:

$$Q = \frac{\sum_{i=1}^{T} V_i}{T} \tag{2}$$

where T is the number of packets in a streaming session and V_i is a variable that takes value 1 if packet i arrives at the receiver before or on its scheduled playout time, and 0 otherwise. The quality is different from throughput because it considers the deadline of each packet. Basically, the parameter Q captures other performance parameters such as packet delay, packet loss and jitter.

Delivery quality can be expressed as a function of contribution, score, or rank. The quality function is system dependent. However, delivery quality should exhibit the following properties : (i) delivery quality is monotonically non-decreasing in user score, (ii) Q_{MAX} represents the highest possible quality provided by the system, (iii) Delivery Quality, Q has non-negative initial value, For example, best-effort delivery quality, $Q_{BestEffort}$=$Q(S_i$=0)≥0.

When a new node first joins the system, it begins with a score of zero and receives best-effort service $Q_{BestEffort}$=$Q(S_i$=0)=0. The quality of this service may vary from system to system, and vary as a function of system load. For example, a supplier node may choose to serve a node through push method with a lower score only when it is idle. Thus, best-effort service quality can be highly unpredictable and often results in lower quality. In order to improve performance and receive better quality than best-effort, a node is required to earn it by contributing to the system and in turn improve its score.

The score is a discrete variable and thus the probability density function (pdf) is defined only where the score has a meaningful value. In order to compute the percentile rank, the cumulative distribution function (cdf) of the scores is calculated. The cdf is defined as:

$$F(S) = \sum_{i=S_{low}}^{S_{high}} f(i) \tag{3}$$

where f is the pdf of the score. The relationship between the percentile rank and the score is provided by cdf. The percentile is obtained by dividing the cdf by the total number of nodes. The scores of all nodes are kept at the streaming server.

We would also like to point out that with systems like PALMS, it is a time sensitive traffic system. Freeriders cannot afford to wait for more time, since each packet has a certain lifetime. In other words, time constrained data distribution provides stronger incentives to peers to discourage the existence of free-riders.

3 PALMS : Packets Scheduling

The algorithms presented in this section make up the core of the PALMS system. They determine how each node chooses its partner for data exchange, how data packets to be sent are chosen and scheduled, which data packets are to be requested from each connected neighbor, and an incentive mechanism to encourage contribution of data received.

3.1 Analysis of Pure Pull Method

We analyzed the detailed process of pure pull method to provide insight into related issue. Basically, the pull component in PALMS is similar to the data-driven approach in DONet. Each node in PALMS periodically exchanges *buffer map* of media packets with neighbors. Based on information gathered from *buffer map*, a node then schedules which packet is to be retrieved from which neighbor accordingly.

In the pull mode of PALMS, when a packet goes from one node to another, the following three steps are executed as shown in Fig. 4. First, the sender receives packets from a connected neighbor and stores them in

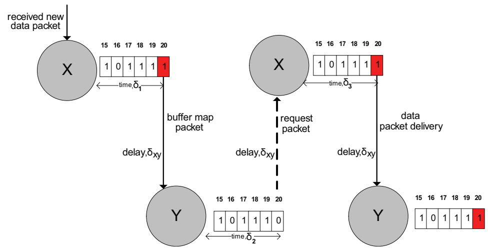


Figure 4: One hop delay using pure pull method

```
Input:
bw[k] : bandwidth from neighbor k
num_suppliers : number of suppliers
bm[i] : buffer map of connected node i
rank[i] : percentile rank of connected node i
deadline[j] : deadline of packet j
expected_set : set of packets to be pulled
set_neighbors : number of neighbors of the node

Scheduling :
  for packet j ∈ expected_set do
      Make num_suppliers = 0
      for l ∈ {1..set_neighbors}
          • Calculate T_j, Time for Transmitting packet j :
          T_j = deadline[j] - current_time
          num_suppliers = num_suppliers + bm[l, i]
      end for
  end for
  if num_supplier=1
      • packets when potential supplier = 1
      for j ∈ {1..expected_set}
      supplier[j] ← arg_r{bm[r, i]=1}
      end for j
  else
      • packets when potential suppliers > 1
      for j ∈ {1..expected_set}
          for r ∈ {1..num_suppliers}
          • Find r with the highest bw[r] and enough
             available time t[r, j]
          supplier[j] ← arg_r{ bw[r] > bw[r'],
                    t[r',j] > t[r,j],rank[j]≤rank[r], r,
                              r' ∈ set_suppliers}
          end for
      end for
   end if
  Output supplier[j]
  Do Pull packets from supplier[j]
  Do Update Buffer Map
```

Figure 5: Pull Method Heuristic Algorithm

buffer. (In this case, the sender is node X while the receiver is node Y). Sender X informs receiver Y about the packets stored in buffer by sending a *buffer map* packet. Second, if receiver Y needs this packet, a request is sent to the sender. Third, sender X will deliver all the requested packets to receiver Y. As depicted in Fig. 4, at least three end-to-end delays are involved in these steps. As a result, the delivery of most packets will have extra delays for a one hop distance. We use δ_1, δ_2 and δ_3 to denote the intermittent waiting time. The total average latency for a packet transmitted in one hop T_{1hop} can be approximately computed as $\delta_1+\delta_2+\delta_3+3\delta_{xy}$, where δ_{xy} is the average end-to-end delay between nodes.

In a nutshell, the pure pull approach displays extra end-to-end latency for packet delivery. In order to improve packet delivery ratio, we propose the combination of push-pull approach. By incorporating the push-pull approach, we expect the following two significant improvements: *(i)* reduce the end-to-end delay observed at the end user node *(ii)* improve the delivery ratio of a media packet to its receiver before the playback deadline.

3.2 Scheduling Algorithm

Given the buffer maps of a node and that of its partners, a schedule is to be generated for the pull and push mechanisms for fetching the expected packets from the partners and sending packets to connected neighbors. Basically, a simple heuristic algorithm is used for both pull and push mechanisms.

3.2.1 Pull Mechanism

The main algorithms used for peer selection for pull and push mechanisms are an altruistic algorithm.

The algorithm for pull methods is similar to the heuristic used in DONet [22] and BitTorrent [8]. The main purpose of the pull method is to request the rarest packets among those that are locally available, and to distribute the request across different possible suppliers. The pull algorithm is shown in Figure 5.

Using the information gathered from the *buffer map* exchanged among neighbor sets, packets that are rarest across the neighborhood are requested with higher priority than more common ones. Packets with the same number of suppliers are randomly requested from one of the neighbors that can provide them. This is to limit the load on any single peer. Based on the rank of individual nodes, a node is only allowed to pull packet from a supplier that has the same or lower score.

3.2.2 Push Mechanism

The pull mechanism is the process of packet delivery by a neighbor after a request is made by a node. Inspired by the work conducted by [4], the push mechanism for PALMS employs two simple techniques too. The push mechanism for PALMS consists of a proactive component where each node pushes data forward, and a reactive mechanism that is triggered by NACKs.

Basically, the push mechanism sends a packet relay to neighbors as soon as the packet is received. Each node works under pure pull mode at the initial stage after joining the streaming network. After that, based on the traffic from each neighbor, the node will subscribe to the pushing packets from its neighbors accordingly at the end of each time interval. Due to the delay that might occur in a pure pull method, a push mechanism helps to increase the delivery ratio of packet to receiver nodes. Moreover, due to the unreliability of the network link or a neighbor failure, some of the packets are lost during transmission. An overlay node can detect missing packet using gaps in the packet sequence numbers. This information is used to trigger NACK-based retransmission through the next interval of push mechanism. Thus, with the help of the push mechanism, packets are pushed and received at the receiver nodes at a second time interval.

A good selection strategy is required to distribute the packets through the push mechanism. This is to

Input:
$set\_neighbors$: number of neighbors of the node
$bm[i]$: buffer map of connected node i
$rank[i]$: percentile rank of connected node k
$deadline[k]$: deadline of packet k
$expected\_set$: set of packets to be pushed

Scheduling :
for packet $k \in expected\_set$ **do**
  for $l \in \{1..set\_neighbors\}$
    • Find Packet with the highest time-stamp :
    T_k = $deadline[k]$ - $current\_time$
  end for
end for
  for $receiver \in \{1..set\_neighbors\}$
• Roulette Wheel Selection for receiver
receiver's with higher $rank[i]$ are given higher probability
  end for
Output $receiver[k]$
Do Push packet to $receiver[k]$

Figure 6: Push Method Algorithm

ensure that every node pushes different packets in order to reduce redundancy. Push packets should also take into account the requests from neighbor nodes. The push algorithm is shown in Figure 6.

For push mechanism packet scheduling, each neighbor node tries to allocate different packets into the *Push Packet Map*, PPm to be pushed. A *Push Packet Map*, PPm consists of node id and packet sequence number. A simple rank based roulette wheel selection scheme is applied for the next time interval for each node to push the available segments. Packets with the highest time-stamp or *least sent* will be given higher priority to be allocated into the *Push Packet Map*, PPm. Each node keeps a counter of how many times each packet is sent. Packets with the least number of times sent will be chosen. In addition to that, packets that required retransmission based on NACKs received will be allocated into the *Push Packet Map*, PPm too.

4 Simulation Scenario

In this section, we perform extensive simulations to study the performance of PALMS. Simulations on the algorithms' behavior test for different network sizes, bandwidth distributions, streaming rates, and number of free-riders using network simulator ns-2 [23].

1) Video Data: The length of the video is 120 minutes (a typical length for a movie).

2) Video Coding: We used a video stream that is MDC encoded with 5 descriptions. For simplicity, we assume that all descriptions have the same constant bit rate of 100 Kbps. Therefore, the rate of the full quality version of the stream is 500 Kbps.

3) Peer Parameters: The incoming access link bandwidth for all peers are set to 500 Kbps. The incoming access links of all peers are set to 500 Kbps so that each peer can easily receive the full quality

Upload B/W	SN 1	SN 2	SN 3	SN 4	SN 5
128 kbps	5%	15%	10%	50%	25%
256 kbps	10%	80%	10%	25%	25%
512 kbps	40%	5%	80%	25%	25%
0 kbps	45%	0%	0%	0%	25%

Table 1: Scenarios for comparing different upload bandwidth under PALMS

playback rate. The buffer length is set to 30 seconds. In all our experiments we use a *heartbeat* period of 5 seconds for all simulated protocols. The interval for the next round of push mechanism is set for every 10 seconds. In order to examine the effects of aggregate available resources, bandwidth heterogeneity and free-riders, the outgoing bandwidth of individual peers can be set to one of four values : 128 Kbps, 256 Kbps, 512 Kbps and 0 Kbps. By controlling the distribution of peers across these four groups, we can control the heterogeneity of outgoing access link bandwidth, the percentage of free-riders that exist in the system i.e., with outgoing bandwidth of zero, which in turn determines the aggregate outgoing bandwidth i.e., system capacity for a given scenario. The distribution of 1000 peers across different groups is shown in Table 1.

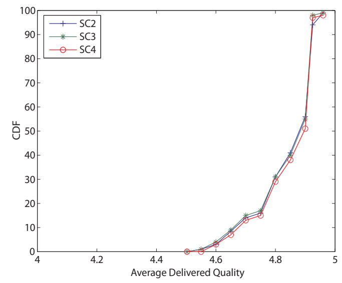
4) Network Topology: Topology is generated by using Georgia Tech Internetwork Topology Models (GT-ITM) generator [21]. The delay on the access links are randomly selected between 5 ms to 25 ms.

5) Performance Metrics: We use three basic Quality of Service (QoS) performance metrics, i.e., Average Delivery Ratio, Delivery Latency and Data Overheads.

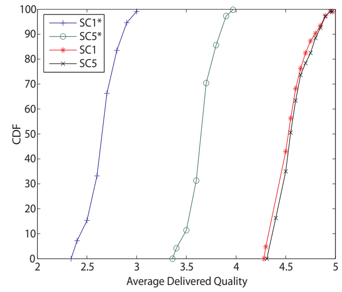
5 Simulation Results

We have examined the impact of heterogenous bandwidth and free-riders on the performance of PALMS streaming. We also study the three metrics of interest: Delivery quality, Delivery latency and Data overheads. We compare the push-pull protocol performance of PALMS with DONet [22] and Chainsaw [16]. Both DONet and Chainsaw employ pure pull mechanism. DONet employs a rarest-first strategy as the block scheduling method, and select suppliers with the most surplus bandwidth and enough available time first. Chainsaw uses a purely random strategy to decide what blocks to request from neighbors.

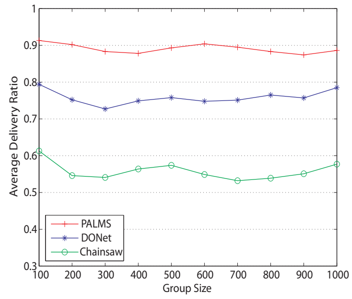
Effects of Heterogenous Bandwidth: We first examine the impact of heterogenous uplink bandwidth on the performance of PALMS. In the experiments, we focus on three scenarios, SN2, SN3, and SN4 as shown in Table 1. We also examine the correlation between the delivered quality (in terms of number of



(a) Effect of Heterogenous Upload Bandwidth

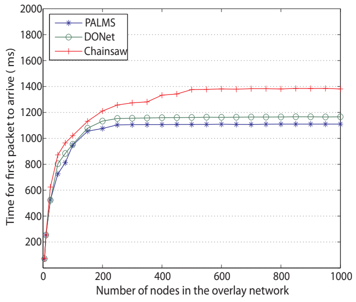


(b) Effect of free-riders on delivered quality

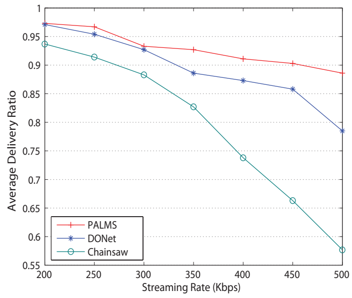


(c) Delivery Ratio as function to Group Size

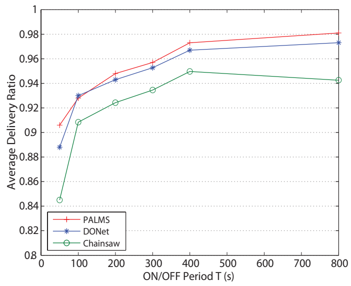
Figure 7: Comparison Simulation Results



(a) Average time for arrival first packet



(b) Average Delivery Ratio as a function to Streaming Rate. Group Size is 1000



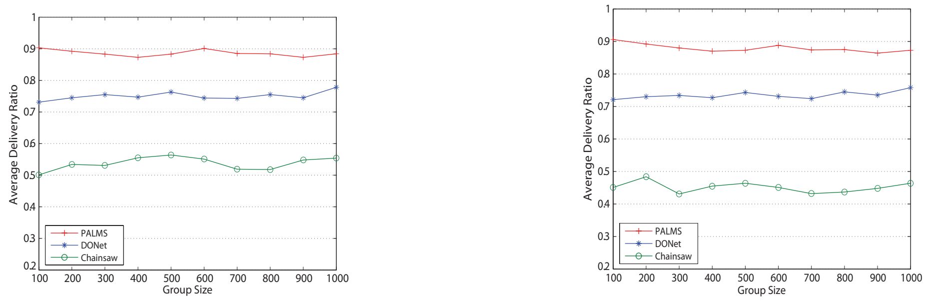
(c) Average Delivery Ratio as a function to ON/OFF Period, T(s). Group Size is 1000

Figure 8: Comparison Simulation Results

description) and contributed resources vis-a-vis outgoing bandwidth of participating peers. Figure 7(a) depicts the CDF of delivered quality and utilization of access upload bandwidth among participating peers for these three scenarios. These figure show that the degree of heterogeneity in upload bandwidth does not affect the distribution of delivered quality of participating peers.

Effect of Free-riders We also investigate the impact of free-riders on the performance of PALMS. For the experiments, we focus on scenarios SN1 and SN5. We set the number of free-riders in the system as roughly 25% and 45% of the total number of participating peers. We examine the performances of PALMS without the implementation of the score-based incentive mechanism. Figure 7(b) shows that the presence of free-riders significantly reduces the delivered quality. We are aware that the scenario with free-riders can be viewed as a special case for bandwidth heterogeneity. Thus, the significant drop in delivered quality as the result of free-riders was rather surprising since the earlier result showed that heterogeneity of bandwidth does not have a major effect on performance.

As we take a closer examination of our results, they revealed that the free-riders affect the connectivity of the overlay. The explanation for such behavior is because free-riders do not have any child peers and their presence in the overlay can affect the connectivity and content exchange between other connected nodes. This in turn limits the delivered quality to other participating nodes. Nevertheless, participating peers are not completely disconnected from the mesh network. The presence of free-riders affects the distribution of content which in turn affects the buffer requirement at each peer. Thus, while a traditional file sharing system can be sustained with low level of cooperation, a P2P streaming system cannot provide high streaming quality to its users if only a small fraction of users cooperate. In short, our results show that the presence of free-riders can significantly affect the connectivity of participating peers in the overlay network which in turn prevents content swarming among peers and thus limits the delivered quality to subset of peers. This shows that there is a need for mechanism to ensure proper connectivity among participat-



(a) Average Delivery Ratio for different group size with 20% free-riders and streaming rate 500kbps. (Group Size 1000)

(b) Average Delivery Ratio for different group size with 50% free-riders and streaming rate 500kbps (Group Size 1000)

Figure 9: Simulation Results on the effect of free-riders

ing peers and reduce the number of free-riders in the network.

Effectiveness of the Incentive Mechanism: We compare the effectiveness of the incentive mechanism for PALMS with the existence of free-riders. Basically, the incentive mechanism provides flexibility to select suppliers from cooperative users to improve the streaming quality. Figure 7(b) shows the system performance under situations with and without the proposed incentive mechanism. Based on the results, most peers have substantially higher quality of media streaming as compared to the system that does not have an incentive mechanism.

Delivery Quality: Figure 7(c) shows the average delivery ratio for PALMS in comparison to DONet and Chainsaw. We define ***delivery ratio*** to represent the number of packets that arrive at each node before playback deadline over the total of number of packets encoded. We set the streaming rate as 500kbps. From the result, we can observe that the performances for PALMS and DONet remain almost the same when group size increases. This is an indication that the performance of swarming based protocols or data-driven protocols is not affected by group size. In other words, swarming protocols have a good scalability. However, Chainsaw method decreases more in comparison to PALMS and DONet. As shown in Fig. 7(c), PALMS has 20% gains compared to DONet and over 45% gains compared to Chainsaw.

Delivery Latency: In Figure 8(a) we show the distribution of latency experienced by data packets at the different overlay nodes. In this experiment, we measure the average time for first packet arrival for all simulated protocols. Note that all protocols suffer an increase in average time of first packet arrival, stabilize, then stay relatively constant with the number of nodes. The increase is well identified and is

Group Size	Control Overheads (Control Traffic/Video Traffic) PALMS	DONet	Delivery Ratio PALMS	DONet
100	0.0173	0.0161	0.91	0.79
200	0.0175	0.0163	0.90	0.75
300	0.0183	0.0171	0.88	0.73
400	0.0184	0.0173	0.88	0.75
500	0.0196	0.0182	0.89	0.76
1000	0.0232	0.0204	0.89	0.79
2000	0.0244	0.0232	0.90	0.79
4000	0.0296	0.0270	0.90	0.80

Table 2: Comparison of Control Overheads for PALMS and DONet

due to the implementation of swarming protocols for PALMS and DONet.

Data Overheads: We compare the overheads of PALMS to DONet to achieve different delivery ratios. Table 2 shows that PALMS incurs very low additional data overheads to achieve relatively high delivery ratios. The control overheads at different overlay nodes increase log-arithmically with the increase in group size. The control overheads for PALMS are higher due to the additional messages such as *Push Packet Map* messages and NACKs. However the amount of increase at each overlay node is essentially minor, less than 3% of the total traffic. We believe the data overheads for PALMS can be further reduced by increasing the window size. We also observe that the data delivery ratio of PALMS is high across various group sizes.

Free-riders: We compare the performance of PALMS, DONet and Chainsaw with the existence of free-riders. We set the number of free-riders as 20% and 50% of the total number of connected nodes. The streaming rate is set as 500kbps. Figure 9(a) and figure 9(b) show the average delivery ratio as a function to group size. As expected, the average delivery ratio for PALMS are significantly better than DONet and Chainsaw for both cases.

Performance under Stable Environment: We examine the performance of PALMS in comparison to DONet and Chainaw under stable environment. We set all the nodes to join in an initialization period of around 1 minute, and then persist in the lifetime of the streaming for 120 minutes, a typical length for a movie. Figure 7(c) and 8(b) show the average delivery ratio as a function to group size and streaming rate. As mentioned earlier, we can see that the average delivery ratio basically remains almost the same when the group sizes increase.

Performance under Dynamic Environment: In Figure 8(c) we show the average delivery quality with dynamic node joining, leaving and failing. Most parameters settings are similar to that in the previous experiment for stable environment. For this experiment, we set each node changes its status according the ON/OFF model. The node actively participates the overlay during the ON period, and leaves (or fails) during the OFF period. Both ON and OFF periods are exponentially distributed. Figure 8(c) shows that a shorter ON/OFF period leads to a lower delivery ratio. However, the overall delivery ratio for PALMS is higher in comparison to DONet and Chainsaw because the additional push mechanism is able to help to recover from a vast majority of losses. Note that Chainsaw displays the poorest performances

These results confirm the expected advantages of the proposed model PALMS for P2P live media streaming.

6 Conclusions and Future Work

In this paper we presented PALMS, a P2P system for live media streaming. Our systems' innovative features are the usage of the combination push-pull protocol and the presence of score-based incentive mechanisms to encourage cooperation among connected nodes.

We also examine the issues and challenges in offering P2P streaming under PALMS. In particular, we focus on the impact of outgoing bandwidth heterogeneity and free-riders on the performance of P2P streaming. We identified that P2P streaming is able to effectively accommodate the heterogeneity of uplink bandwidth, but the presence of free-riders could significantly affect the connectivity of the overlay and reduce the feasibility of data exchange among nodes. This result significantly degrades the delivered quality to a subset of peers in the system.

To successfully deploy PALMS streaming services, we proposed push-pull score-based incentive mechanism to address the issue of delivery quality, delivery latency and free-riders. We conducted simulations and we showed that a push-pull score-based incentive mechanism achieves cooperation through service differentiation. In this framework, the contribution of a user is converted into a score and mapped into a rank, and the rank provides flexibility in peer selection that determines the quality of a streaming session. Cooperative users earn higher rank by contributing their resources to others, and eventually receive high quality of streaming. Free-riders have limited choice in peer selection, hence receive low quality streaming.

We evaluated the performance of PALMS in comparison to DONet and Chainsaw. Our simulations conducted over ns2 demonstrated that PALMS delivers quite a good playback quality even under formidable network conditions and the existence of free-riders.

As part of our future plans, we aim to evaluate our proposed model, PALMS in PlanetLab [24], in order to to further investigate the effectiveness and the robustness of our streaming model in a larger network and real network deployment. We are also keen in exploring various techniques to improve the delivery quality and delivery latency.

Acknowledgments

This work was supported by the by the Ministry of Education, Culture, Sports, Science and Technology, Government of Japan, and also by a grant from the Hori Foundation Science Promotion Foundation, Japan.

References:

[1] E. Adar and B.A. Huberman, "Free Riding on Gnutella," First Monday, 5(10), October, 2000.

[2] S. Agarwal and S. Dube, "Gossip Based Streaming with Incentives for Peer Collaboration," ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia, pp.629–636, December, 2006.

[3] Akamai Technologies Inc, "Delivery a better Internet," [Online] Available: http://www.akamai.com/.

[4] S. Banerjee, S. Lee, B. Bhattacharjee and A. Srinivasan, "Resilient multicast using overlays," IEEE/ACM Transactions on Networking, pp. 237–248, vol. 14, no. 2, 2006.

[5] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application layer multicast," SIGCOMM '02: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols

for Computer Communications, pp. 205–217, August, 2002.

[6] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream : High-Bandwidth Multicast in Cooperative Environments," SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, pp. 298–313, Bolton Landing, NY. USA, October 2003.

[7] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," Measurement and Modeling of Computer Systems, pp. 1–12, June 2000.

[8] B. Cohen, "Incentives build robustness in BitTorrent," P2P Economics Workshop, Berkeley, CA, 2003.

[9] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming live media over peers," Tech. Rep. 2001-31, CS Dept, Stanford University, 2001.

[10] Duc A. Tran, Kien A. Hua, and Tai T, Do, "A Peer-to-Peer Architecture for Media Streaming." IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, pp. 121–133, January, 2004.

[11] Y. Guo, K. Suh, and D. Towsley, "A Peer-to-Peer on-Demand Streaming Service and Its Performance Evaluation," ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo, pp. 649–652, Washington, DC, 2003.

[12] A. Habib and J. Chuang, "Incentive mechanism for peer-to-peer media streaming," IWQOS 2004: 12th International Workshop on Quality of Service, pp. 171–180, Montreal, Canada, 2004.

[13] M. Hefeeda, A. Habib, R. Rotev, D. Xu, and B. Bhargava, "PROMISE: peer-to-peer media streaming using CollectCast," MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia, pp. 45–54, Berkeley, CA, November, 2003.

[14] D. Kostic, A Rodrigues, J. Albrecht, and A. Vahdat, "Bullet: high bandwidth data dissemination using an overlay mesh," SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, pp. 282–297, Bolton Landing, NY, October, 2003.

[15] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient Peer-to-Peer Streaming," ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols, Atlanta, GA, pp. 16–27, November, 2003.

[16] Pai, V. Kumar, K. Tamilmani, K. Sambamurthy, V. Mohr, A. E., "Chainsaw: Eliminating Trees from Overlay Multicast," IPTPS, no. 3640, pp. 127-140, 2005.

[17] S. Saroui, P.K. Gummadi, and S.D Gribble, "A measurement study of peer-to-peer file sharing systems," Proceedings of Multimedia Computing and Networking, 2002.

[18] J. Song, C. Sha, and H. Zhu, "Nash Equilibria in Parallel Downloading with Multiple Clients," ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04), pp. 94–101, 2004.

[19] Q.Sun and H. Garcia-Molina, "SLIC : A selfish link based incentive mechanism for unstructured peer-to-peer network," Distributed Computing Systems, pp. 506–515, 2004.

[20] S. Ye and F. Makedon, "Collaboration-aware peer-to-peer media streaming," MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia, pp. 412–415, NY, 2004.

[21] E. W. Zegura, GT-ITM: Georgia Tech Internetwork topology models (software). http://www.cc.gatech.edu/project, 1996.

[22] X. Zhang, J. Liu, B. Li, and T.P. Yum, "CoolStreaming/DONet : A data-driven overlay network for efficient live media streaming," INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 2102–2111, March, 2005.

[23] Network Simulator, Ns-2. http://www.isi.edu/nsnam/ns/

[24] The PlanetLab project, http://www.planet-lab.org/