EXPERIENCES WITH A LARGE-SCALE DEPLOYMENT OF THE STANFORD PEER-TO-PEER MULTICAST

Jeonghun Noh, Pierpaolo Baccichet†, Bernd Girod*

Information Systems Laboratory, Department of Electrical Engineering
Stanford University, Stanford, CA, 94305, USA
Email: jhnoh@stanford.edu, pier@dyyno.com, bgirod@stanford.edu

ABSTRACT

Traditionally, a large number of dedicated media servers have been deployed to serve a large population of viewers for a single streaming event. However, maintaining media servers is not only costly but also usually requires over-provisioning due to the difficulty of predicting the peak size of an audience. Peer-to-Peer (P2P) streaming is a new approach to overcome these difficulties inherent in server-based streaming. We have developed the Stanford Peer-to-Peer Multicast (SPPM) protocol for live multicast streaming. SPPM constructs multiple multicast trees to push media streams to the population of peers, thereby achieving low end-to-end transmission delay. The degradation of video quality due to peer churn and packet loss in the network is reduced by video-aware packet scheduling and retransmission.

In this paper, we present lessons we acquired from the deployment of a commercial variant of SPPM for a large-scale streaming event which attracted more than 33,000 viewers. We collected server logs and analyzed user statistics as well as the system performance. The results show that our system can achieve low end-to-end delay of only a few seconds with an average packet loss ratio of around 1%. We also found that improving peer-to-peer connectivity can substantially enhance the aggregate uplink capacity of P2P systems.

Index Terms— Peer-to-peer multicast streaming, P2P, large-scale deployment, low-latency streaming.

1. INTRODUCTION

Recently, live multicast of social events on the Internet has gained much popularity. Unlike regular TV programs, social events including live concerts, presidential debates, and technical conferences, tend to take place intermittently, and yet attract a large population of users in single gatherings. Traditionally, a large number of dedicated media servers have been deployed to serve such big flash crowds. However, maintaining media servers for one-time events is not only expensive but also often requires over-provisioning to lower the number of rejected user requests. Peer-to-Peer (P2P) streaming has received much attention as a promising and cost-effective solution for live multicast [1–3]. P2P systems improve their own streaming capacity by harnessing the uplink bandwidths of participating peers. This can reduce or eliminate the needs for media servers.

We developed the Stanford Peer-to-Peer Multicast (SPPM) [4–8] protocol as a robust and low-latency P2P streaming system. SPPM constructs multiple multicast trees to push media streams to the population of peers, thereby achieving low end-to-end transmission delay. The degradation of video quality due to peer churn and packet loss in the network is reduced by video-aware packet scheduling and retransmission. In this paper, we present our experiences with deploying a commercial variant of SPPM for the ESWC, a large-scale streaming event[1]. To offer users a quality video, SPPM was enhanced by Dyyno Inc. [10] with a hybrid P2P concept of deploying super nodes. During this event, about 33,580 viewers from 93 countries watched the live multicast. We collected the empirical data including user behavior, user characteristics, and the session logs for evaluating system performance. The remainder of the paper is organized as follows. Section 2 describes the original SPPM protocol. Section 3.1 provides an overview of the system used in the experiments. In Section 3.2, we describe the experiments and methods for data collection. In Section 4, the analysis of the experimental results is presented.

2. STANFORD PEER-TO-PEER MULTICAST

The SPPM protocol organizes peers in an overlay of multicast trees. Every multicast tree is rooted at the video source. The source packetizes the video stream and distributes video packets to each multicast tree in a round robin manner. Peers subscribe to all multicast trees in order to receive the video stream contiguously.

*This work was performed when Jeonghun Noh was an intern at Dyyno Inc, which kindly provided the ESWC logs used in this paper. Dyyno Inc. is located at 2100 Geng Road, Suite 103, Palo Alto, CA, 94301, USA.

†Pierpaolo Baccichet is now with Dyyno Inc.

[1]ESWC, standing for Electronic Sports World Cup, is an international professional gaming championship [9].

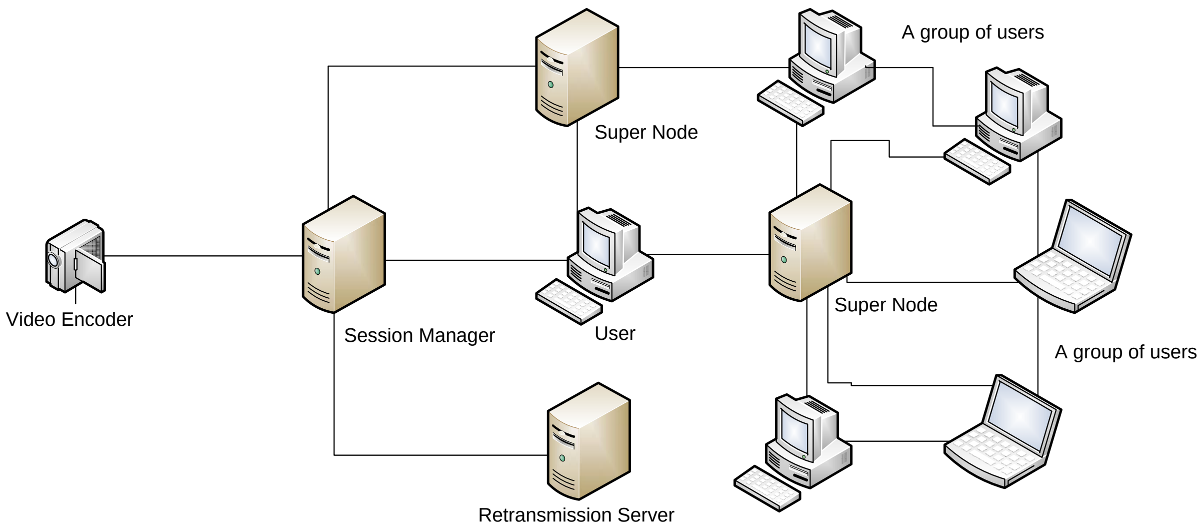

Fig. 1. An example system configuration: video is captured and encoded in real time. A session manager receives the encoded signal and delivers it to peers. Peers consist of a set of super nodes and a group of users.

2.1. Overlay construction

The overlay is incrementally constructed as new peers join the system asynchronously. A newly joining peer first contacts the video source. The video source replies with session information, such as the number of multicast trees and the video bitrate. It also sends a list of parent candidates that are randomly chosen among peers already connected to the system. When the new peer probes the parent candidates, they report their current status including available bandwidth and depth in the multicast trees. The new peer then selects the best parent candidate for each tree in such a way that the height of each multicast tree is minimized. Once the selected parents accept the connection request from the new peer, data connections are established between the parents and the peer.

2.2. Overlay management

After data transmission starts, each child peer periodically sends *Hello* messages to their parents. When a peer leaves the system ungracefully, its parents detect it from the lack of consecutive *Hello* messages and stop forwarding video to the child. The children of the departing peer (their parent) detect it from the lack of video packets as well as the lack of response to their *Hello* messages. Each disconnected child then initiates a recovery procedure to connect to a different peer in the tree. For missing packets due to network congestion or parent failure, retransmission requests are made according to the packets' predicted impact on video playback [7]. When a peer generates a retransmission request, it sends the request to one of its parents which it randomly chooses.

3. EXPERIMENTS

3.1. System configuration

Fig. 1 illustrates an overview of the live streaming system deployed on the Internet. For video delivery, the system employs a variant of SPPM, which is enhanced for commercial grade streaming service. The system consists of a real-time video encoder, a session manager, a retransmission server, and a population of peers. Peers comprise super nodes and a group of users. Super nodes are special servers that can provide additional relaying capacity to the system. From the SPPM protocol's point of view, they act as normal peers with high uplink bandwidth, which are willing to relay video to other peers. The session manager constantly monitors the status of a session, including the aggregate of the uplink bandwidths. When the aggregate bandwidth is not sufficient to support an entire peer population, the super nodes provide their uplink bandwidths to the system. The bandwidth provided by super nodes can be written as $NR - \alpha U$, where NR is the required bandwidth (N: total number of peers, R: video bitrate), and U is the aggregate of the uplink bandwidths of the session manager and the users. $\alpha \in (0, 1)$ is a system parameter for compensating for the inaccuracy of the uplink bandwidth measurement and the limited peer-to-peer connectivity due to NAT and firewall. Addition or termination of super nodes during a session does not affect the system operation because the protocol is designed to tolerate unpredictable peer behavior. Adding servers to a P2P system is a popular solution to quality video streaming, also found in other P2P systems [11–14].

During a live session, a video signal is encoded in real

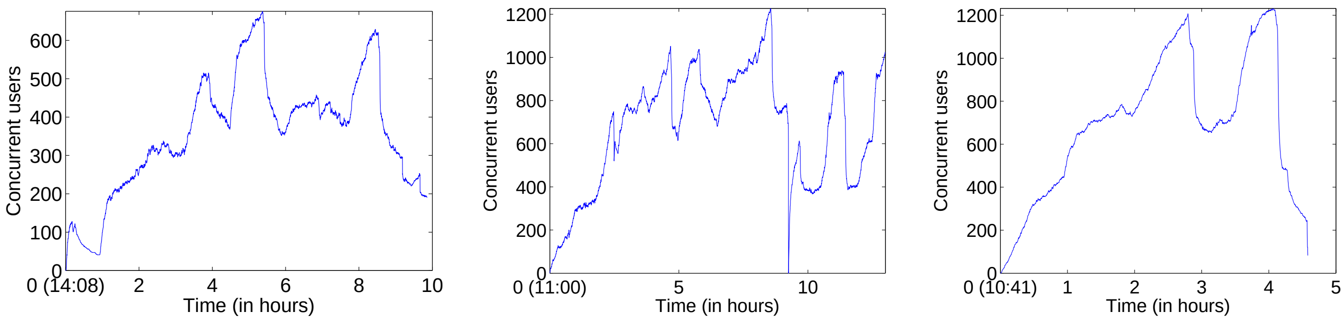

Fig. 2. Number of concurrent users. Times are relative hours to the starting time. The actual starting time is specified next to Hour 0. Left: Day 1. Middle: Day 2. Right: Day 3

time at the venue of the event. The encoded video stream is transmitted to the session manager. The session manager acts as a video source in the multicast session as if it were the origin of the video. For video distribution, eight multiple trees are built in a distributed manner. Super nodes are usually found near the root of each multicast tree as the system promotes peers with high uplink bandwidth, called *high-profile peers*, by moving them toward the tree root.

To serve missing or late packets, a retransmission server is placed in each session. Each peer closely checks the timestamps of received video packets and generates retransmission requests if it detects missing packets that are shortly due for playout. In the original design of SPPM, retransmissions were handled only locally among adjacent peers in the overlay. Retransmission servers complement the local retransmission by providing reliable retransmissions of video packets that are close to their playout deadline. Video-aware packet prioritization and retransmission algorithm are performed as in the original SPPM.

In our system, users join the system using their personal computers. An SPPM client is installed as a web browser plug-in when a user visits a well-known web portal (a so-called rendezvous point). All users watch the same video scene simultaneously[2]. To provide high interactivity to the users, the maximum end-to-end transmission latency is on the order of few seconds (less than 5 seconds).

System robustness can be assured by adding redundancy to the system. In our system, this redundancy is achieved by setting up a number of identical sessions. Each session is assigned a session manager and a retransmission server, and serves the identical content.

3.2. Data collection

We collected data from the ESWC 2008, held in San Jose, California, USA, from the 25th to the 27th of August [9]. The event was a world-wide PC game tournament hosted annually by ESWC. It was broadcast live using the system described in Section 3.1. A variety of online PC game matches were broadcast for 5 to 12 hours every day. Each match lasted around 30 minutes to 1 hour. The video was encoded using H.264/AVC at 10 frames per second with an intraframe period of 20 frames. To avoid additional encoding latency, B frames were not employed. The spatial resolution was 640 by 480 pixels. No protection, such as FEC, was added to the bitstream. Instead, robustness to packet loss and/or peer churn was provided by retransmission. A mono channel of the audio was sampled at 44.1 kHz, 16 bits/sample and compressed using AAC. The total bitrate was approximately 600 kbps (the video accounted for 560 kbps and the audio for 40 kbps). The VLC player [15] was used for video playback at each peer. In the VLC player, when packets were late or missing, the last correctly decoded frame was displayed until the next I frame arrived. Most of the video scenes were computer-generated, fast-moving 3D graphics. In general, the visual quality was acceptable when the packet loss ratio (PLR)[3] was less than 2%.

Peers reported basic statistics, such as PLR and connection duration to the session manager at regular intervals. Retransmitted packets were considered to be received on time as long as they arrived before their deadline. We collected peer reports and the general session information generated by the session managers, including the size of the peer population, session length, and the average PLR.

4. ANALYSIS

We analyze the data obtained from ESWC 2008 below. Individual statistics for each day are presented since averaging across sessions might not highlight peculiarities of each session. Fig. 2 shows the number of concurrent users who watched the event each day. The peaks in the graphs were

[2] A chat room was provided to allow users to share their opinions. To this end, TV-like concurrent watching of a program was desired.

[3] PLR is the ratio of the number of missing/late video packets to the total number of video packets.

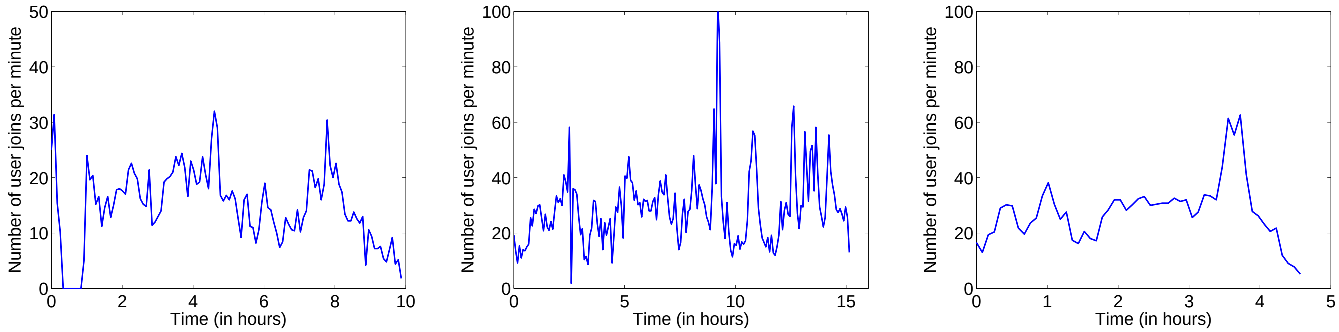

Fig. 3. User join rate (averaged over a 5-minute sliding window). Left: Day 1. Middle: Day 2. Right: Day 3

usually followed by a sudden drop due to the simultaneous departure of users who, most likely, had finished watching the matches they were interested in. On Day 2, there was a brief system outage around Hour 9. Around this time, the number of connected users dropped to nearly 0. After the system became available shortly, about half of the disconnected users rejoined the system immediately in the next few minutes.

The user join rates are displayed in Fig. 3. The join rate at each moment was obtained by averaging over a 5-minute-long sliding window. There were intermittent high join rates during the event, particularly near the beginning point of new matches. The join rate oscillated around the day average join rate as opposed to exhibiting a monotonous increase or decrease over time (see Fig.3). Except at some moments of the event at which people showed up as a flash crowd, the join rate remained relatively stable throughout the event. Table 1 shows the average join rate on each day of the event.

Date	User join rate per minute
Day 1 (August, 25^{th})	14.9
Day 2 (August, 26^{th})	29.2
Day 3 (August, 27^{th})	27.8

Table 1. Average number of users arriving at the system per minute.

The distribution of peer uplink bandwidth is shown in Fig. 4. More than half of the users contributed less than 600 kbps of uplink bandwidth, where 600 kbps was the total streaming bitrate. Since each multicast tree delivered a fraction of the stream, peers with uplink bandwidth lower than the total bitrate were still able to contribute to the aggregate system capacity. The average uplink bandwidth was 1647 kbps, much higher than the video bitrate contributed by high-profile peers. As other researchers have pointed out [12], an effective treatment of high-profile peers is critical in improving the relaying capacity of the system.

Next, we study the accessibility of peers, which affects peer-to-peer connectivity in a practical P2P system. When

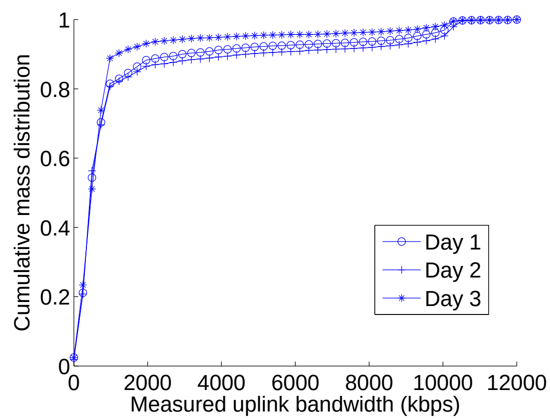

Fig. 4. Peer uplink BW distribution.

an SPPM client joined the system, it discovered the existence of a NAT (or a firewall), and the type of NAT by running a variant of the STUN algorithm [16]. We classified peers according to four different NAT-type categories – *full cone*, *restricted IP*, *restricted port*, and *symmetric NAT* – and according to two non-NAT type categories, *static IP* and *static IP firewall*. Fig. 5 depicts the distribution of the NAT type of peers. We observed no distinct difference across the days, indicating that there is a weak stationarity in the distribution of peer NAT types. The majority of peers turned out to be restricted-port NAT type, accounting for half of the peer population. In Fig. 6, the peer contributions of uplink bandwidth are depicted according to their NAT type. It shows that the contribution ratio from peers without NAT is higher than the ratio of the number of corresponding peers to the population. A plausible explanation is that a large number of peers without NAT are part of a high-bandwidth network, such as a university campus or research institution.

NATs usually block connection attempts from outside the network. Peers with static IP addresses, yet behind a firewall, are not visible to the outside unless they first initiate communication with other peers. Peers with full-cone NAT are

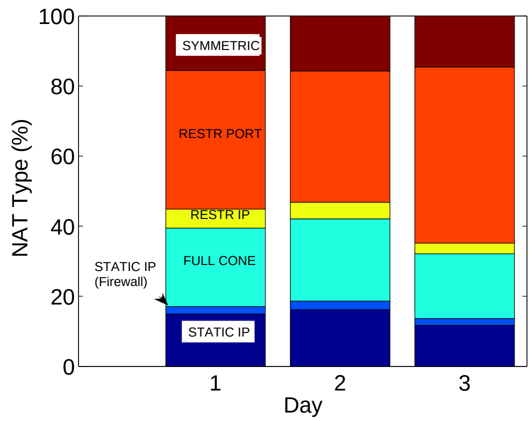

Fig. 5. NAT type of peers for each day. (Symmetric: symmetric NAT. RESTR PORT: Restricted Port NAT. RESTR IP: Restricted IP NAT. FULL CONE: Full cone NAT)

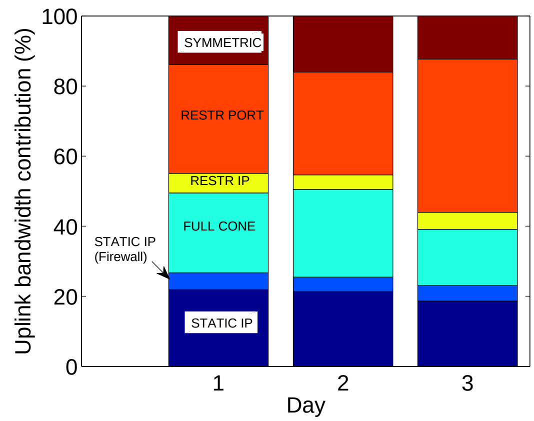

Fig. 6. Peer contribution for each connectivity type. (Symmetric: symmetric NAT. RESTR PORT: Restricted Port NAT. RESTR IP: Restricted IP NAT. FULL CONE: Full cone NAT)

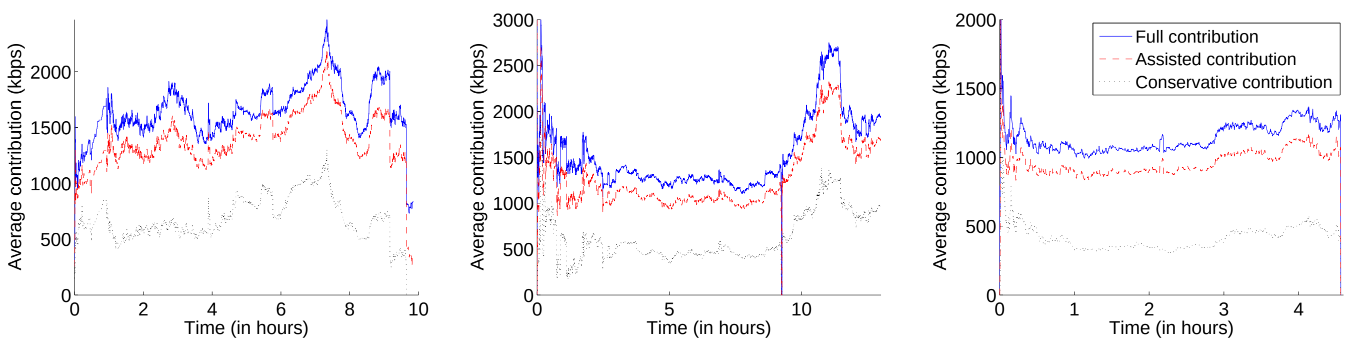

Fig. 7. Temporal evolution of the average uplink bandwidth. Left: Day 1, Middle: Day 2, Right: Day 3.

accessible from outside once their public address assigned at the NAT device is known. Peers with static IP and peers with full-cone NAT are most easily accessible in a P2P system. Peers with restricted-IP NAT accept traffic only from the network addresses (regardless of transport layer port number) with which they have initiated communication. Thus, both peers with restricted-IP NAT and peers with static IP, yet behind firewalls become accessible to external peers only if they first initiate communication with them. Peers with restricted-port NAT only accept external traffic with a network address and a transport layer port number with which they have initiated communication. Peers with symmetric NAT are assigned a different pair of address and port for each connection they establish with external computers. Peers in this NAT category have the most limited accessibility. These limitations in peer-to-peer connectivity due to NATs and firewalls can pose a serious difficulty in overlay construction [17]. With external assistance, such as UDP hole punching [18], peer-to-peer connectivity can be substantially improved, thus increasing the aggregate uplink capacity of a P2P system. Table 2 lists peer-to-peer connectivity between different NAT types of peers with the help of the UDP hole punching technique.

To see the impact of external assistance on peer-to-peer connectivity, we consider the following three contribution scenarios:

- Conservative contribution: average uplink bandwidth contributed by a peer when no external assistance is provided. Only static IP and full-cone NAT peers can contribute their bandwidth without external assistance.

- Assisted contribution: average uplink bandwidth contributed by a peer if external assistance is employed. Besides static IP and full-cone NAT peers, peers behind the other types of NATs or firewalls can contribute their bandwidth by employing UDP hole punching [18]. However, the hole punching technique fails to provide connectivity between some pairs of NAT types, as marked in Table 2.

- Full contribution: average uplink bandwidth con-

	Static IP	Full cone	Restricted IP	Static IP Firewall	Restricted Port	Symmetric
Static IP	○	○	△	△	△	△
Full cone	○	○	△	△	△	△
Restricted IP	○	○	△	△	△	△
Static IP Firewall	○	○	△	△	△	△
Restricted Port	○	○	△	△	△	×
Symmetric	○	○	△	△	×	×

Table 2. Connectivity chart. Column: connection initiator (source). Row: connection acceptor (destination). ○: Always possible. △: Possible with external assistance. ×: Not possible.

tributed by a peer in case of no restriction in peer-to-peer connectivity. It provides the upper bound of peer contribution.

These peer contributions are expressed in (1) and the notations are listed in Table 3.

$$
\begin{aligned}
N_{all} &= \sum_{\forall type} N_{type} \\
BW_{all} &= \sum_{\forall type} BW_{type} \\
C_{con} &= \frac{BW_{ST} + BW_{FC}}{N_{all}} \\
C_{ast} &= \Big\{ BW_{RI} + BW_{SF} + BW_{RP} \cdot (1 - \frac{N_{SM}}{N_{all}}) + \\
&\quad BW_{SM} \cdot (1 - \frac{N_{SM} + N_{RP}}{N_{all}}) \Big\} \cdot \frac{1}{N_{all}} + C_{con} \\
C_{full} &= \frac{BW_{all}}{N_{all}}. \qquad (1)
\end{aligned}
$$

Symbol	Definition
N	Number of peers
BW	Total uplink bandwidth
C_{con}	Conservative contribution
C_{ast}	Assisted contribution
C_{full}	Full contribution
ST	Static IP
SF	Static IP behind firewall
FC	Full-cone NAT
RI	Restricted-IP NAT
RP	Restricted-port NAT
SM	Symmetric-NAT

Table 3. Symbols used in (1)

Fig. 7 illustrates the average uplink bandwidth over time according to the different contribution scenarios. Overall, the assisted contribution was nearly 85% of the full contribution. This observation clearly emphasizes the benefit of external assistance in improving peer-to-peer connectivity. On Day 1 and 2, the fluctuation of the average uplink bandwidth was significant during the event. The high fluctuation and the uncertainty in peer contribution often results in a P2P system being forced to serve users with a less than ideal video quality. If the system is required to serve a consistently high quality of video, additional relaying capacity must be provided by dedicated relay servers, such as super nodes.

Fig. 8 graphs the probability mass function of the user connection durations. About half of the users left the system within five minutes after they joined. In Fig. 9, the data and the two model distributions are plotted together. The exponential distribution is popular in simulations and the mathematical analysis of P2P system performance due to its simplicity and special properties, such as memorylessness. In addition, we chose the Zipf distribution, known for its long tail[4]. The two model distributions were fitted to minimize the squared errors between the model and the empirical data. In Fig. 8, the fitted exponential distribution model had a mean of 8.0 minutes, which was much smaller than the actual average of 27.1 minutes. The fitted Zipf distribution had a parameter of $s = 1.59$, which is expressed as

$$f(X = x) = \frac{1/x^{1.59}}{\sum_{n=1}^{N}(1/n^{1.59})}, \qquad (2)$$

where X takes on the medium value of each 5-minute-long interval. As pointed out in prior work [11, 12], the user connection duration follows the Zipf distribution closely. Another implication of the Zipf distribution model is the correlation between user connection duration and lifetime. We considered the conditional probability that a peer may leave the system during the next 30 minutes[5] given that the peer has been present in the system for t minutes. Fig. 10 illustrates the conditional probability in (3) for the data and two models.

$$\Pr(T > t + 30 | T = t), \qquad (3)$$

where T denotes a peer's connection duration and both T and t are in minutes. The memoryless property of the exponential

[4] Since the distribution based on the empirical data was represented as a PMF, we selected the Zipf distribution in lieu of the Weibull distribution.

[5] A large period (of 30 minutes) was chosen to smooth out the curve obtained from the empirical data.

	Full contribution	Assisted contribution	Conservative contribution
Day 1	1647 kbps	1385 kbps	672 kbps
Day 2	1530 kbps	1283 kbps	627 kbps
Day 3	1141 kbps	959 kbps	414 kbps

Table 4. Average uplink bandwidth contributed by a peer based on contribution type.

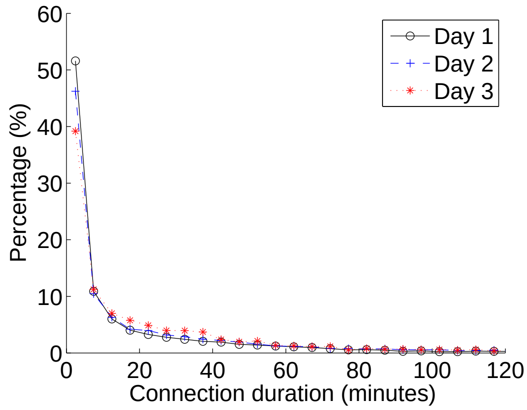

Fig. 8. The probability mass function (PMF) of connection durations.

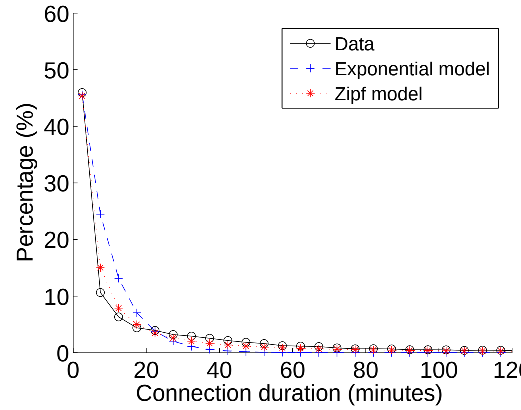

Fig. 9. The PMFs of connection durations. Two model distributions are fitted against the real data.

distribution implies that a peer's departure probability during the next 30 minutes is 97%, regardless of its connection duration. However, the data show the actual departure probability was much lower than 97% and invalidate the memoryless property with a monotonous decrease over time. The Zipf distribution model shows a similar trend as the data. However, the Zipf model proved to predict a lower peer departure rate than the data. The lesson we learned from the correlation between connection time and lifetime can be reflected in the construction of overlays; since peers who have recently joined the system have a high departure probability, young peers should not be promoted toward the tree root too early. When peers look for parents, they may prefer as parents peers who have been in the system longer.

In Fig. 11, we presented the geographical distribution of users[6]. All connections were counted separately based on their IP address and port combination; if the same user connected to the system several times on a day, then these connections were counted separately. Users from 93 countries participated in the event online. The distribution of user's geographic location reflects the *Pareto principle (the 80/20 rule)* that 84.7% of the users come from 20% of the countries (18 countries). The fact that the majority of users are located closely to each other show that P2P system designers should take user proximity into consideration in designing a P2P system [20].

[6]"the Country WhoIS DB" (July 10, 2008 release) available from TamoSoft [19] was used to map an IP address to its geographical location.

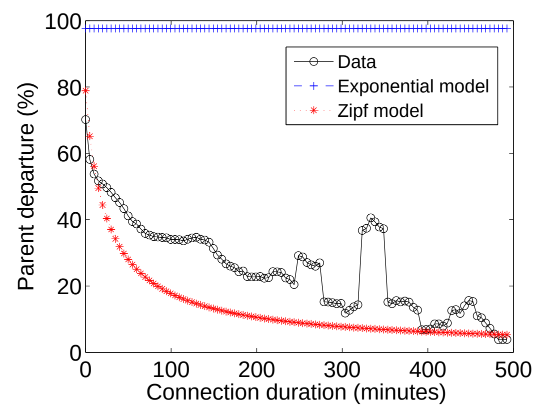

Fig. 10. Conditional distribution of the departure of an active peer within the next 30 minutes.

Fig. 12 shows the packet loss ratio (PLR) distribution according to the user's geographic locations. From the 93 countries, we selected 35 countries in which more than 100 users watched the event. We grouped the 35 countries according to their continents. The individual continent PDFs showed similar patterns, with a peak occurring near 0.5% PLR. This result implies that peers' physical locations are not highly correlated with streaming quality.

Fig. 13 displays the PLR against user uplink bandwidth. This indicates that no strong correlation exists between user uplink bandwidth and PLR. Since the system treats peers

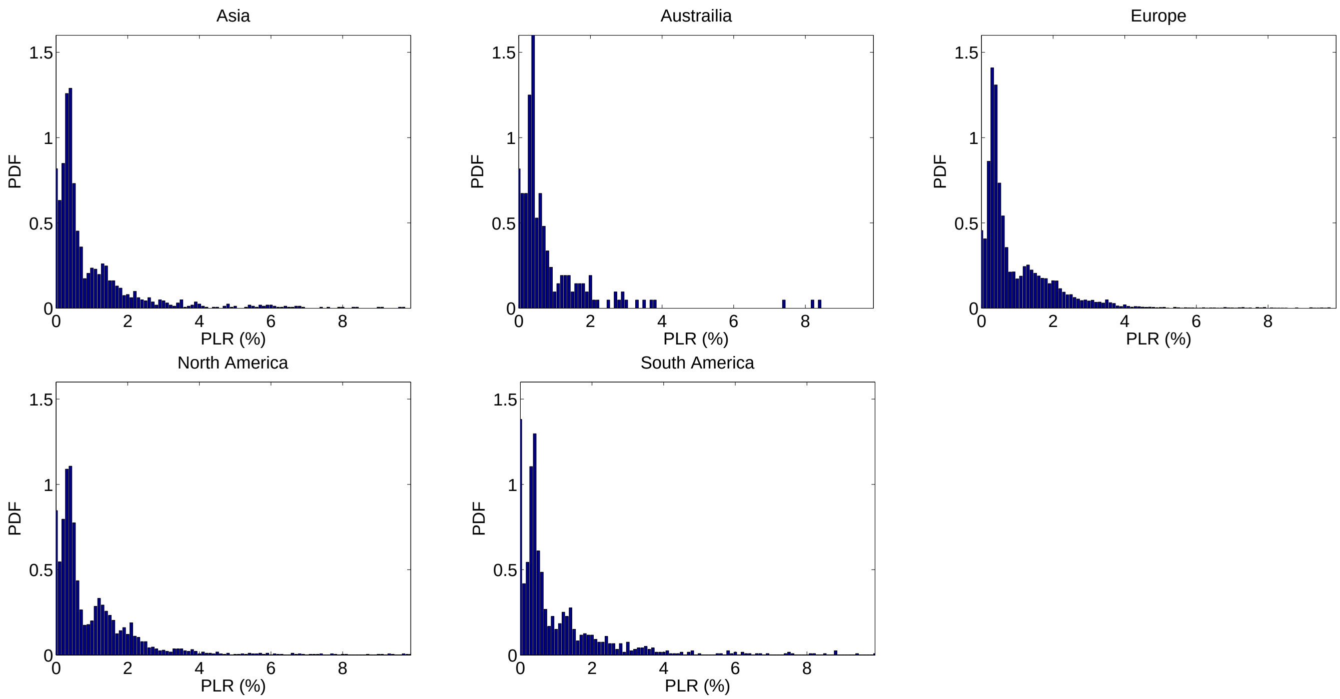

Fig. 12. PLR against geographical locations (all days). The countries in which more than 100 users watched the video were grouped together into their continents.

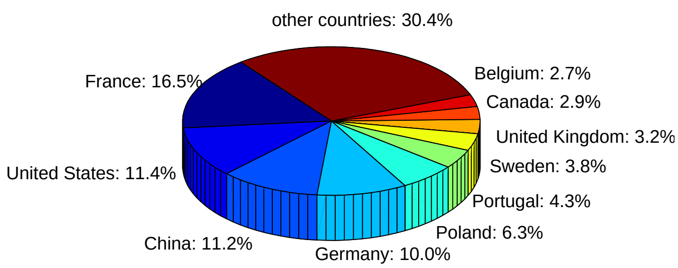

Fig. 11. User's geographic distribution (all days). Users came from 93 countries in all.

equally regardless of their net contribution to the system, peers experience the same quality of video as long as their downlink bandwidth is higher than the video bitrate.

Fig. 14 depicts the PLR evolution over time. The three-day average PLR was 1.34%. Some peaks occurred intermittently, possibly due to the transient rearrangement of the overlay structure. This indicates that retransmission servers can fail to serve retransmission requests in a timely manner. On Day 2, around Hour 9, there was a short system outage. On Day 3, the average PLR was below 0.5% most of the time (the daily average PLR was 0.52%). When the number of concurrent users changed rapidly, a temporarily high PLR was observed. This result suggests that the system is less susceptible to the total number of users than to the fluctuation in the user population, such as peer arrival and departure.

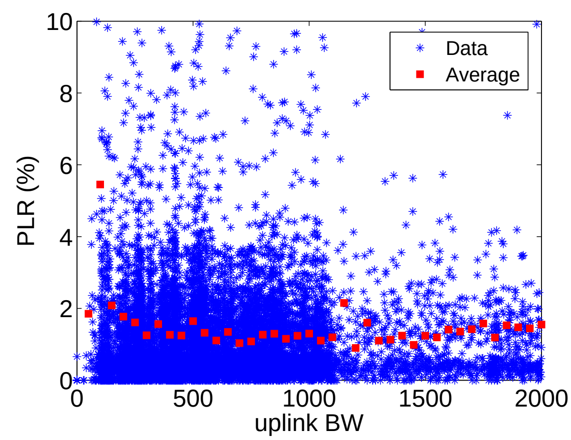

Fig. 13. PLR (packet loss ratio) against user uplink BW.

5. CONCLUSIONS

One of the difficulties in building a peer-to-peer system is limited knowledge about peer characteristics. In this paper, we presented a rich set of peer characteristics and the performance of the SPPM protocol based on a large-scale video session. Our analysis shows that the system can achieve low-latency end-to-end transmission delay (less than a few seconds) as well as good video quality; the all-day average packet loss ratio (PLR) was 1.34% and the Day 3 average

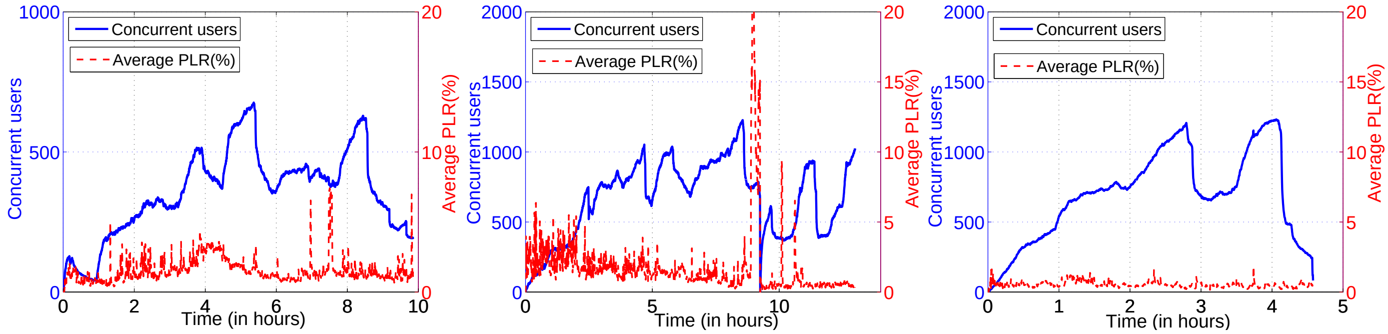

Fig. 14. Temporal evolution of the PLR and concurrent users. Left: Day 1, Middle: Day 2, Right: Day 3.

PLR was 0.52% for a session of 1200 users. We also showed that peers bring sufficient uplink bandwidth into the system if peer-to-peer connectivity is improved with external assistance. The analysis also revealed that the aggregate of peers' uplink bandwidths fluctuated widely during the session. In the experiments, super nodes offered additional relaying capacity to the system when the aggregate uplink capacity was insufficient.

6. REFERENCES

[1] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," *Proc. of ACM NOSSDAV, Miami Beach, FL*, May 2002.

[2] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming," *Proc. of IEEE Infocom, Miami, USA*, Feb. 2005.

[3] Y. Chu, A. Ganjam, T. Ng, S. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an internet broadcast system based on overlay multicast," *USENIX*, Nov. 2004.

[4] E. Setton, J. Noh, and B. Girod, "Rate-Distortion Optimized Video Peer-to-Peer Multicast Streaming," *Workshop on Advances in Peer-to-Peer Multimedia Streaming at ACM Multimedia, Singapore*, pp. 39–48, Nov. 2005.

[5] E. Setton, J. Noh, and B. Girod, "Low-latency video streaming over peer-to-peer networks," *Proc. of International Conference on Multimedia and Expo, Toronto, Canada*, pp. 569 – 572, Jul. 2006.

[6] E. Setton, J. Noh, and B. Girod, "Congestion-distortion optimized peer-to-peer video streaming," *Proc. of International Conference on Image Processing, Atlanta, Georgia*, pp. 721 – 724, Oct. 2006.

[7] E. Setton, P. Baccichet, and B. Girod, "Peer-to-Peer live multicast: A video perspective," vol. 96, no. 1, pp. 25–38, 2008.

[8] P. Baccichet, J. Noh, E. Setton, and B. Girod, "Content-Aware P2P Video Streaming with Low Latency," *Proc. of IEEE Int. Conference on Multimedia and Expo, Beijing, China*, Jul. 2007.

[9] "ESWC San Jose 2008," *http://www.eswc.com /grandfinal/sanjose2008/*.

[10] "Dyyno inc.," *http://www.dyyno.com/*.

[11] Y. Chu, A. Ganjam, T.S.E. Ng, S.G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an internet broadcast system based on overlay multicast," *Proc. of USENIX*, Jun. 2004.

[12] B. Li, S. Xie, G. Y. Keung, J. Liu, I. Stoica, H. Zhang, and X. Zhang, "An Empirical Study of the Coolstreaming+ System," *IEEE Journal on Selected Areas in Communications*, pp. 1627–1639, Dec. 2007.

[13] M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang, "Large-scale live media streaming over peer-to-peer networks through global internet," *Proc. of ACM Int. Conference on Multimedia, P2PMMS Workshop*, pp. 21–28, Nov. 2005.

[14] E. Setton and J. Apostolopoulos, "Towards quality of service for peer-to-peer video multicast," in *Proc. of IEEE International Conference on Image Processing*, Sep. 2007.

[15] "the VLC player," *http://www.videolan.org/vlc/*.

[16] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) through Network Address Translators (NATs), RFC 3489," Mar. 2003.

[17] A. Ganjam and H. Zhang, "Connectivity restrictions in overlay multicast," *Proc. of the 14th international workshop on Network and operating systems support for digital audio and video*, pp. 54 – 59, Nov. 2004.

[18] B. Ford, D. Kegel, and P. Srisuresh, "Peer-to-peer communication across network address translators," in *Proc. of the 2005 USENIX Technical Conference*, 2005.

[19] "TamoSoft," *http://www.tamos.com/*.

[20] R. Zhang and Y. Charlie Hu, "Anycast in locality-aware peer-to-peer overlay networks," in *Proc. of the 5th International Workshop on Networked Group Communication*, 2003.