TOWARDS QUALITY OF SERVICE FOR PEER-TO-PEER VIDEO MULTICAST

Eric Setton and John Apostolopoulos

Streaming Media Systems Group
Hewlett-Packard Labs, Palo Alto, CA
{eric.setton, john_apostolopoulos}@hp.com

ABSTRACT

Peer-to-peer streaming is a novel, low-cost, paradigm for large-scale video multicast. Viewers contribute their resources to an overlay network to act as relays for a real-time media stream. Early implementations fall short of the requirements of major content owners in terms of quality, reliability, and latency. In this work we show how adding a limited number of servers to a peer-to-peer streaming network can be used to enhance performance while preserving most of the benefits in terms of bandwidth cost savings. We present a theoretical model which is useful to estimate the number of servers needed to ensure fast connection times and improved error resilience. Experimental results show the proposed approach achieves 10x to 100x bandwidth cost savings compared to a content delivery network, and similar performance in terms of quality and startup latency.

1. INTRODUCTION

In live peer-to-peer (P2P) video multicast, a stream is transmitted to a large audience, utilizing the uplink bandwidth of participating peers. Similar to popular file transfer networks, such as BitTorrent, media delivery is accomplished via a distributed protocol which lets peers self-organize, for example, into application-layer multicast trees [1, 2, 3]. The striking difference is that data transfer happens in real-time, to provide all connected users with a synchronous, TV-like, viewing experience. Compared to content delivery networks (CDNs), this type of distribution system is appealing as it does not require any dedicated infrastructure and is scalable as the resources of the network increase with the number of users.

For content owners, P2P streaming networks would enable large-scale media distribution on the Internet, an application which is not profitable today, because the cost of bandwidth exceeds that of advertisement revenues. However, these customers require a product which achieves the same performance as a CDN, i.e., high and reliable quality, and low startup latencies. Such quality requirements are very difficult to achieve with a P2P-based approach. Unlike client-server systems, a new viewer needs to locate other peers that have sufficient available throughput to act as relays, before it can establish a connection to the P2P network. This causes additional startup latency. Moreover, any peer may log off from the system at any time, thereby disrupting media stream distribution to a potentially large fraction of the network.

The purpose of this work is to show that these difficulties may be overcome and that the performance of P2P streaming systems can be enhanced, provided a limited number of servers are used for error-resilience and connection assistance. As servers would be needed, in any case, for authentication and license management, we believe this is a small price to pay to make P2P networks competitive with CDNs. This paper proposes a hybrid approach, where servers are used to improve the performance of P2P video streaming systems. We focus particularly on the cases where servers forward the stream to new viewers, until these users establish their connection to the P2P network, and where the servers act as fallbacks when a peer is partially disconnected from the network. In the next section, we describe server-assisted P2P streaming in more detail, while keeping a general perspective as we believe our approach is applicable to most P2P streaming systems. We show how to model such a system and evaluate its bandwidth requirements. In Section 3, we analyze experimental results obtained over a simulated network with thousands of peers.

2. SERVER-ASSISTED P2P STREAMING

Figure 1 depicts what we denote by “server-assisted P2P video streaming” or “hybrid P2P streaming”. As illustrated, joining peers first connect to a CDN (on the left of the figure), from which they transition to the P2P network (on the right of the figure), as they establish connections to other peers which will act as relays for the media stream. This transfer is shown by the flow denoted by R. When a client of the P2P network logs off, the transmission of the media stream to its direct descendants will be disrupted. In this case, these peers transfer to the CDN while connection to the P2P network is re-established. This transfer is shown by the flow denoted by Q. As servers provide reliable error resilience to these clients, subsequent peers which they serve in the P2P network will not be affected by loss and therefore, the disruption caused by the client that logged off is minimized.

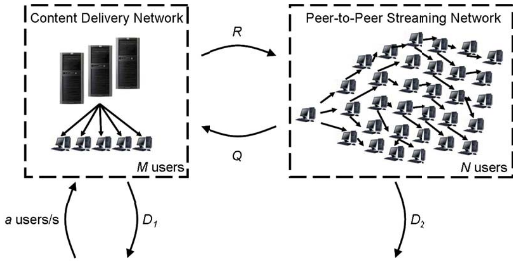


Fig. 1. Server-assisted P2P video streaming model.

2.1. Theoretical Model

The number of users connected to the CDN and to the P2P network are respectively denoted by M and N. We assume, for simplicity, that the lifetime of a user on the system is exponentially distributed, with parameter λ, and that the amount of time needed to connect to the P2P system is also exponentially distributed, with parameter μ. In addition, we assume these random variables are independent, stationary, and that they do not depend on the state of the system. This is the case for a scalable P2P protocol which maintains good performance even for large number of users, e.g., PPLive [4], Gridmedia [5], which have both reported over 100,000 simultaneous viewers. We denote by aT the number of users joining the system in an interval of time T and by D_1 and D_2 the number of users leaving the system during this interval, depending on whether the users are connected to the CDN or to the P2P network. The number of users transferring between the CDN and the P2P network, during this time, are denoted by R and Q.

The total number of users in the system is straightforward to compute. As a consequence of assuming an exponentially distributed lifetime, the expected number of users leaving the system is:

$$E[D_1 + D_2] = (M + N)(1 - e^{-\lambda T}) \tag{1}$$

In steady state, it is equal to the number of arrivals aT. Therefore, the total number of users in the system when an equilibrium is reached is:

$$M + N = \frac{a}{\lambda} \tag{2}$$

which is obtained by taking the limit for $T \to 0$.

In the following we show how to compute the number of users that transfer between the CDN and the P2P network, Q and R, as functions of λ and μ. We introduce the following indicator functions which simplify the derivation:

- $X_k, k \in \{1 \cdots M\}$, denotes whether User k in the CDN stays logged on, in the next interval of length T.
- $Y_k, k \in \{1 \cdots N\}$, denotes whether User k in the P2P network stays logged on, in the same interval.
- $J_k, k \in \{1 \cdots M\}$, denotes whether User k transitions from the CDN to the P2P network, in the same interval.
- $C_{ki}, k \in \{1 \cdots N\}$, denotes whether User k forwards the stream to User i. Please note that $C_{ii} = 0$

The number of users that transfer between the CDN and the P2P network are:

$$\begin{aligned} R &= \sum_{k=1}^{M} X_k J_k && (3) \\ Q &= \sum_{k=1}^{N} (1 - Y_k) \sum_{i=1}^{N} C_{ki} Y_i && (4) \end{aligned}$$

Equation (4) can be interpreted as follows. When a user of the P2P network logs off, the peers it was forwarding data to connect to the CDN to keep receiving the media stream. This happens while they reestablish their connection to the P2P network, and only if they are still logged on (hence the presence of the term Y_i in (4)). This interpretation holds for single multicast tree systems where a peer is receiving the data stream from a single parent. It is identical for a multiple multicast tree system, or for a data-driven system such as Gridmedia [5], if we consider C_{ki} to be fractional (e.g., $C_{ki} = 0.5$ would imply that User k forwards half of the media stream to User i). This more general interpretation does not affect the following derivation.

As we assume an exponentially distributed lifetime and connection time on the CDN, the expected value of (3) is:

$$\begin{aligned} E[R] &= E[\sum_{k=1}^{M} X_k J_k] && (5) \\ &= \sum_{k=1}^{M} E[X_k J_k] = \sum_{k=1}^{M} E[X_k] E[J_k] && (6) \\ &= e^{-\lambda T}(1 - e^{-\mu T}) M && (7) \end{aligned}$$

The derivation in (6) is a consequence of the independence of X_k and J_k. The expected value of (4) is:

$$\begin{aligned} E[Q] &= \sum_{k=1}^{N} E[(1 - Y_k) \sum_{i=1}^{N} C_{ki} Y_i] && (8) \\ &= \sum_{k=1}^{N} E[(1 - Y_k)] \sum_{i=1, i \neq k}^{N} E[C_{ki} Y_i] && (9) \\ &= \sum_{k=1}^{N} E[(1 - Y_k)] \sum_{i=1, i \neq k}^{N} E[C_{ki}] E[Y_i] && (10) \\ &= e^{-\lambda T}(1 - e^{-\lambda T}) \sum_{k=1}^{N} \sum_{i=1, i \neq k}^{N} C_{ki} && (11) \\ &\simeq e^{-\lambda T}(1 - e^{-\lambda T}) N. && (12) \end{aligned}$$

In (9)-(11), we use the independence between the different Y_k and Y_i (for $k \neq i$) and between Y_k and C_{ki}, and $C_{ii} = 0$. In

(12), we observe that the sum of the children of all the peers is simply the total number of peers. The sign $\simeq$ is used as, strictly speaking, the source of the multicast also forwards the stream to some peers which are not counted in this sum. However, for a large network their number represents a negligible amount.

In steady-state, $E[R] = E[Q] + E[D_2]$. By taking the limit for $T \rightarrow 0$, we obtain:

$$N = \frac{\mu M}{2\lambda}. \tag{13}$$

Combining (2) and (13), we can express N and M as:

$$N = \frac{\mu}{\lambda(2\lambda + \mu)} a, \quad M = \frac{2}{(2\lambda + \mu)} a. \tag{14}$$

2.2. Discussion

The expressions derived in (14) allow to compute the expected behavior of a server-assisted P2P video streaming system provided the lifetime of a peer on the system can be estimated, as well as the expected join time of the P2P control protocol, and the expected number of users joining the system in steady-state. In addition to its mean, given by (14), it is interesting to analyze the dynamic behavior of the system, illustrated in Fig. 2. The parameter values are given in the caption. The lifetime of the peers and the join time of the P2P control protocol are assumed to be exponential.

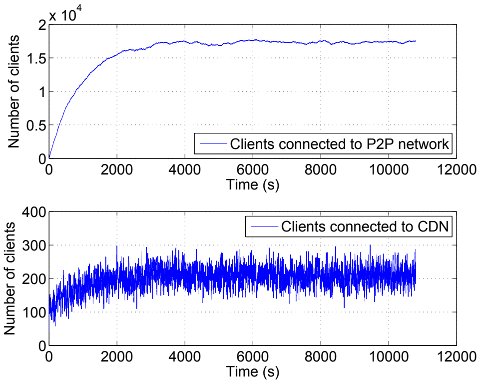


Fig. 2. Dynamic behavior of a server-assisted P2P system. The number of peers joining per second is $a = 20$, the expected lifetime of a user is 15 minutes, and the joining latency of the P2P network is 5 seconds.

At equilibrium, the system holds a total of 18,000 peers, only 200 of which are connected to the CDN, as predicted by the model. As illustrated, the variance of the number of users connected to the CDN, M, is limited. For these parameters, over-provisioning the CDN so that it could support 300 users would be sufficient. Compared to a traditional CDN system, the bandwidth cost savings accomplished by such a system would be a factor of 60. More generally, for meaningful values of λ and μ the gains range between 10 and 100x. We stress, that performance of the hybrid P2P streaming system would be comparable to that of a CDN: the joining time would be similar, as users would initially connect to the CDN; in addition, when P2P distribution fails, due to the dynamic behavior of the clients, error resilience is provided by the CDN. Besides, it would be possible to achieve any given performance between that of a pure CDN and that of a pure P2P network by limiting the resources of the CDN to different levels.

Our simulated model is also useful to study flash crowds or massive disconnection events which are known to happen frequently and have a large impact on the performance of P2P networks. Due to space limitations we cannot fully analyze this effect in this paper. We briefly mention, however, that flash crowds and massive disconnections would place a higher burden on the CDN. As these events are punctual and often depend on content stream, statistical multiplexing would offset this additional cost for a system designed to serve a number of different streams.

3. EXPERIMENTAL RESULTS

In this section, we analyze results obtained over a realistic hybrid P2P streaming network, simulated in NS-2. In these experiments thousands of peers run a distributed P2P control protocol which was designed at Stanford University (see, e.g., [6]) and deployed recently in PlanetLab. The protocol has been modified to support hybrid P2P streaming, where a CDN is used during startup and disconnections, as described and modeled in Sec. 2.

3.1. Simulation setup

Simulations are run over a network topology with a few thousand nodes. The actual number of peers participating in each run varies between 1500 and 2000. The backbone links are sufficiently provisioned so that congestion only occurs on the links connecting the peers to the network. The latency of each link is 5 ms, and the diameter of the network is 10 hops. Losses are only due to congestion and queue overflow, and transmission errors due to the presence of ISP boundaries or potential wireless last-hop links are ignored. The bandwidth distribution for the access links reflects today's popular network access technology; it is given in [6], where the protocol is also described in more detail. The uplink and downlink of the CDN, which is also placed at the edge of the network is assumed to be 40 Mb/s.

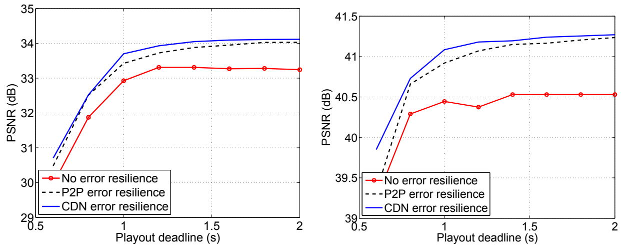
A 10s video sequence is transmitted from the source to the peers along 4 multicast trees. It is looped enough times to simulate a 30 minute session. The video stream is encoded with H.264 at a constant quality and the encoding rate is approximately 300 kb/s. Each video frame is packetized into

UDP packets. Performance is collected for different playout deadlines, i.e., the time between when a packet is available at the source and its decoding deadline at the peers. When a video packet arrives after its playout deadline, the picture is frozen until the next decodable frame.

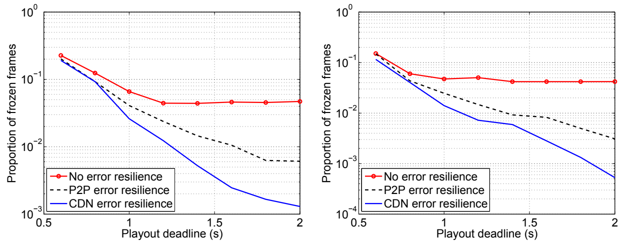
In the experiments, peers frequently log “on” and “off”, and their dynamic behavior is modeled as in [6]. In particular, they stay connected to the system for an average time of 4.5 minutes (i.e., $\lambda = 1/270$, using the notation of the previous section), and there is approximately 1 join/s. The join latency of the protocol is around 1 s (i.e., $\mu = 1$).

3.2. Performance Analysis

To illustrate the performance of our hybrid approach, we compare its error-resilience to that of a P2P-based technique where partially disconnected peers request the missing portion of the video from other connected peers, as they try to reconnect to the network. This technique is described in detail in [6]. As these peers do not necessarily receive all the packets, the performance is expected to be lower than for the CDN-based approach. We also report the performance when there is no error resilience at all. Results for two popular sequences are shown in Fig. 3, in terms of decoded quality, measured as the Peak Signal to Noise Ratio (PSNR) of the luminance component of the video signal, and in terms of the fraction of frozen frames, averaged over the 300 peers.



(a) Foreman - video quality (b) M. & D. - video quality



(c) Foreman - losses (d) M. & D. - losses

Fig. 3. Results for the sequences *Foreman* (left) and *Mother and Daughter* (right) encoded at 290kb/s and 282 kb/s.

As illustrated in Fig. 3, there is a 0.5-1 dB visual quality improvement when error resilience is enabled. In this case, the performance improves with longer playout deadlines, as there is more time to recover from a disconnection. The video quality of the hybrid approach exceeds that of the pure P2P approach by up to 0.3 dB, a modest gain. More importantly, the fraction of frozen frames is significantly reduced. For playout deadlines of over 1.5 s it is always below 0.5% for the hybrid approach, 4 to 5 times lower than the pure P2P approach. This is significant as this kind of reliability would be necessary for mainstream adoption of P2P-based streaming.

The rate served by the CDN, for peers which are partially disconnected of the network, varies between 1 Mb/s and 1.4 Mb/s, on average. This corresponds to supporting between 3 and 4 users. This is close to what our model predicts, in steady state. Evaluating (14), for the parameters given above, leads to $N = 2$ users.

The latency of the two systems (pure P2P and hybrid P2P) can be compared by examining the time at which the first picture is shown in both systems. For the pure P2P, the first picture is played on average after 1.8 s (with a standard deviation, $stdev = 0.7\ s$). In the hybrid approach the average time is 0.6 s ($stdev = 0.2\ s$). In other words the latency is reduced by a factor of 3. This also is significant, and would make a clear difference to a user switching between channels.

4. CONCLUSIONS

This paper introduces “server-assisted P2P streaming” where a P2P network is combined with a CDN to enhance performance in terms of latency and error resilience. We analyze this system both theoretically and experimentally. Compared to a CDN, and for typical parameters, bandwidth cost savings range between 10x and 100x. Our experiments show that compared to a state-of-the-art P2P streaming system, the number of frozen pictures is reduced by a factor of 5 and the startup latency by a factor of 3. We believe these significant results could boost the adoption of P2P-based video multicast.

5. REFERENCES

[1] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable Application Layer Multicast,” *Proc. ACM SIGCOMM, Pittsburgh, USA*, pp. 205–217, Aug. 2002.

[2] D. Tran, K. Hua, and T. Do, “ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming,” *Proc. IEEE INFOCOM, San Francisco, USA*, vol. 2, pp. 1283–1292, Mar. 2003.

[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,” *Proc. USENIX Annual Technical Conference, Boston, MA, USA*, pp. 1283–1292, June 2004.

[4] “PPLive,” *http://www.PPLive.com, seen on Aug. 28 2005.*

[5] M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang, “A Peer-to-Peer Network for Live Media Streaming using a Push-Pull Approach,” *Proc. ACM Int. Conference on Multimedia*, pp. 287–290, Nov. 2005.

[6] 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*, pp. 39–48, nov 2005.