MULTI-VIEW VIDEO STREAMING OVER P2P NETWORKS WITH LOW START-UP DELAY

Engin Kurutepe, Thomas Sikora

Communication Systems Group, TU Berlin,
EN-1, Einsteinufer 17, 10587 Berlin, Germany
{kurutepe,sikora}@nue.tu-berlin.de

ABSTRACT

We propose to stream multi-view video over a multi-tree peer-to-peer (P2P) network using the NUEPMuT protocol. Each view of the multi-view video is streamed over an independent P2P streaming tree and each peer only contributes upload capacity in a single tree, in order to limit the adverse effects of ungraceful peer departures. Additionally, we introduce a quick join procedure to reduce the start-up delay for the first data packet after a join request. Continuity index and decoded video quality performance for simulcast and MVC encoding in a large topology under different settings are reported, in addition to the improvements achieved by the quick join procedure.

***Index Terms*—** Multi-View Video, Peer-to-peer networks, Video Streaming

1. INTRODUCTION

There is a growing interest towards 3-D entertainment applications from both content providers and end users. It is being expected that about 50% of all theaters will be 3-D capable by the end of 2009. Given the interest in 3-D, it will be unsurprising to see 3-D content migrate from theaters to home entertainment systems, just like movies migrated from movie theaters to our TV-sets. Therefore, there is a need for new technologies, which would enable transmission of new 3-D content to end users. In line with the convergence of telephony and TV communications to IP-networks, the authors expect IP-only delivery of 3-D content in the near future.

Multi-view representations provide a good compromise between the amount of data and rendering complexity and they can be compressed using simulcast encoding, where each view is encoded as an independent conventional video stream. However, more efficient compression can be achieved by exploiting the spatial redundancy between cameras, in addition to the temporal redundancy in a camera stream [1]. This approach is currently being standardized under JVT as an amendment to H.264/AVC [2]. However, even after state of the art compression, multi-view representations are very data intensive.

High bit-rate of multi-view representations is an important challenge for both the content providers and end users. At the end user side, there has been work on selective streaming [3], such that only the necessary parts of the multi-view video are streamed according to the viewer's head position. This kind of selective streaming systems would benefit greatly from scalability features of the MVV stream which have been proposed in [4] and [5]. On the other hand the problem at the server-side is that given a fixed bandwidth the number of viewers which can be served at the same time is reduced drastically due to much increased bitrate.

There has been a wealth of research on Peer-to-Peer (P2P) video streaming, and it has been shown that P2P architectures can greatly increase the number of simultaneously served viewers. The underlying idea beneath the P2P architectures is that the peers share their disk and bandwidth resources with the P2P network by intelligently forwarding the packets they receive, to other peers. This allows for the bandwidth load to be lifted from the server and distributed among peers. Although P2P streaming has been shown to be feasible [6] and cost-effective [7], there are two important intrinsic issues facing any P2P approach: Reliability of end systems and upload/download bandwidth asymmetry for most peers. Reliability of end systems is inherently lower than dedicated routers and when an end system ungracefully leaves the network, all its children are starved until a new parent is found. Upload/Download bandwidth asymmetry affects how the total capacity of the P2P network grows as new peers join the network. As many of the domestic cable and DSL connections offer less bandwidth for upload than for download, many peers can lack the upload bandwidth to contribute into the network, even if they have enough download bandwidth to watch the content, thus becoming free-riders in the network. It has been shown [8] [9] [10] that multiple description coding (MDC) can help for both issues. MDC involves dividing a video stream into several independent descriptions, which are then streamed over different paths in a P2P setting, making use of path diversity to ensure at least some of the descriptions arrive at their destinations in the case of peer departures. Additionally since each of these descriptions has a lower bi-

This work is supported by EC within FP6 by 3DTV Network of Excellence under Grant 511568.

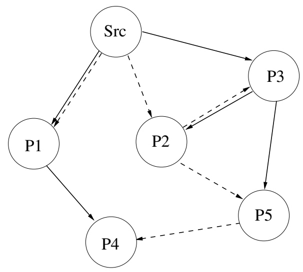

Fig. 1. A simple multi-tree network with two independent trees, where each peer contributes capacity to a single tree.

trate in comparison to the original stream, a peer which has not enough upload bandwidth for the original stream, could contribute capacity in the P2P network for some of the MDC descriptions.

The parallelism between multi-view video and multiple description coding of conventional video is noteworthy. Whereas separate descriptions are streamed over separate trees in P2P streaming of MDC encoded video, we propose to consider each view in an MVC encoded multi-view video as a separate description and stream each view over an independent P2P streaming tree to get similar benefits as in conventional video streaming. In this paper we will present how packet losses due to ungraceful peer departures affect the continuity of the playback at the viewers and the decoded video quality. We also introduce a quick join procedure, which allows new peers to start receiving packets before they have properly joined the tree.

The rest of this paper is organized as follows: In Section 2 we describe the details of the proposed approach. Section 3 defines the experimental setup and presents our results. And finally in Section 4 we list our conclusion and outlook for the future.

2. SYSTEM DESCRIPTION

In this section, we describe a simple multi-tree P2P live streaming protocol which was used to examine how P2P streaming works for multi-view videos. But before we proceed to the details of our protocol, the interaction between different multi-view encoding schemes and multi-tree streaming should be discussed properly.

Simulcast encoding regards each view as an independent video stream and encodes using H.264/AVC. Obviously, the effects of packet losses in simulcast encoded streams are contained within the same stream and do not propagate to neighboring views. Additionally, we also consider two flavors of multi-view coding (MVC). First one with complex spatial prediction structure where there are spatial reference between all neighboring frames, and the second simplified prediction structure, where there are only spatial references between frames at anchor time instances at the beginning of each group of pictures (GOP). It was shown in [11] that this simplified prediction structure enables much less complex decoding and simplified random access within the stream with a negligible loss of compression efficiency. Unlike simulcast encoding the effects of packet losses are likely to affect neighboring views in both MVC prediction schemes. The losses propagate to neighboring views when an anchor frame is lost with the simplified scheme, whereas almost every packet loss affects neighboring views with the complex scheme.

2.1. The P2P protocol: NUEPMuT

NUEPMuT stands for NUE[1] P2P Multi-Tree. The tree management is currently handled by the source node, since the management processing and bandwidth overheads are negligible beside the load of multi-view streaming.

2.1.1. Join Process

A new peers joins the network using the six-way handshake process: first a JOIN packet containing the total maximum capacity the joining peer is willing to contribute to the network is sent to the source node. The source replies for each tree with a list of fertile peers and indicates if capacity is requested in that particular tree. The source allocates the total capacity of the new peer to the tree, which currently has the minimum total available capacity. Therefore, each peer is fertile only in a single tree and acts as a free-rider in the other trees as seen Fig 1. This has been shown to increase stability [6] by making sure that an ungraceful peer departure disrupts only a single tree. Then the new peers sends PING packets to each peer in the lists provided by the source, which in turn reply with their depth in the tree and current available capacity. Finally, the new peer sends ATTACH packets to the peers with minimum tree depth in each tree, where the peer with the lowest RTT measurement is preferred between peers with the same depth. If the ATTACH request is rejected, the new peer selects the next suitable candidate for that tree, or retries to join that tree again if no other candidates are available. An ATTACH acknowledgement also contains the list of the ancestors for loop prevention.

2.1.2. Tree Management

NUEPMuT has a heartbeat frequency, with which each node sends out a small packet to its parent. Therefore, a parent can detect dead children which do not send their heartbeat packets. Conversely, a parent replies to each heartbeat packet it receives from its children. A parent which does not reply to a heartbeat packet is assumed to be dead and triggers a new join process. During these rejoins, attach requests are denied if a rejoining peers is in the ancestor list of a potential parent.

[1]NUE is the german abbreviation for Communication Systems Group

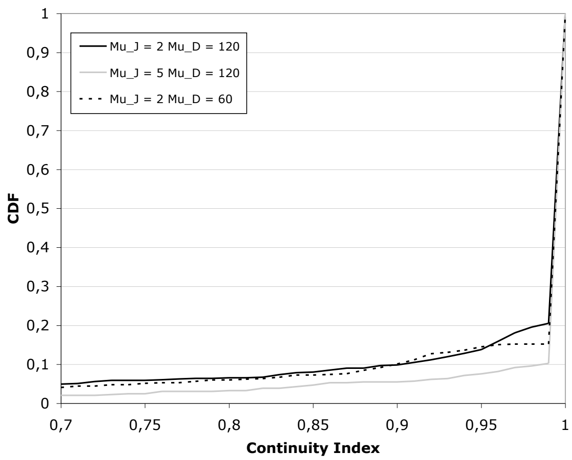

Fig. 2. Cumulative distribution of packet continuity index for three different simulation scenarios.

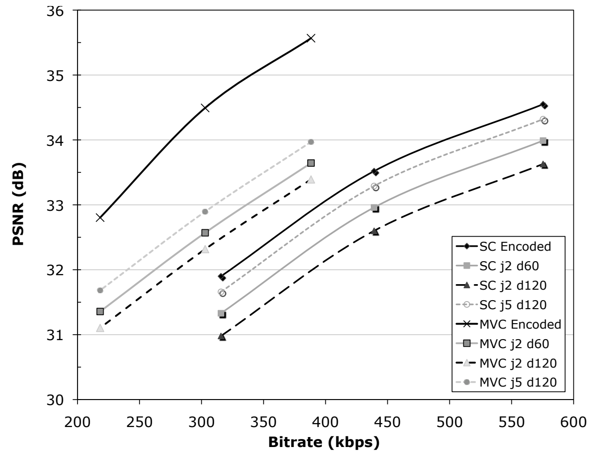

Fig. 3. Rate-distortion curves averaged over all peers for simulcast and MVC at different peer-churn rates

Additionally, peers inform the source when they don't have available capacity left due to a new join, or when capacity becomes available due to graceful or ungraceful departures. This information is used in the join process to select the tree with minimum total available capacity and to maintain a list of fertile peers.

2.1.3. Quick Join

The six-way handshake employed during the join process can take some time, especially in networks which span a large geographical area. In order to enable the joining peers to quickly start receiving packets, the source instructs the peer with the maximum available capacity in each tree to start forwarding packets to the joining peer. This is a temporary data forwarding is cancelled if the temporary parent has not enough capacity left due to new children attaching to it or as soon as the new peer has properly joined the tree and starts receiving packets form its proper parent.

3. RESULTS

We have implemented the NUEPMuT protocol in NS-2 network simulator. The worst case scenario with only ungraceful departures was assumed. This is not strictly realistic, as in a real-life scenario it is reasonable to expect a significant number of viewers to depart gracefully. However, the assumption of ungraceful departures provides a lower-bound on the achievable performance.

The proposed P2P streaming protocol was simulated on a 400-node network, with over-provisioned link bandwidths to filter out the effects of congestion. Nevertheless, each node contributes a randomly determined amount of upload resources to reflect the effects of a realistic upload bandwidth distribution as presented in [6]. The outgoing bandwidth of the server was fixed at a relatively low 5Mbps and the total bitrate of the encoded MVV was 1Mbps with 256Kbps per view. Under this topology settings the peers joined the network with exponentially distributed inter-arrival times with mean μ_J and stayed in the network for an exponentially distributed duration with mean μ_D. The first four views of the KDDI_Race1 sequence were encoded JM10.2 and JMVM3.0 for simulcast and multi-view respectively. IDR frames were inserted at the beginning of each GOP for both encoding schemes. The simplified MVC structure [11] is used for MVC encoded video.

Fig. 2 shows the cumulative distribution of continuity index for delivered packets under various μ_J and μ_D values. The general observation here is, that the μ parameters have relatively little effect on the packet continuity and under all three simulation settings the protocol was able to provide the peers, with a good continuity index with most of the peers enjoying 100% packet continuity. However, it should be noted that this good continuity comes at the cost of new peer admission. During the simulations about 25-30% of new join requests were rejected, due to insufficient capacity in the network. On the other hand, the number of users which can be simultaneously served with the given server bandwidth still increased about four to five fold in comparison with a unicast system.

The average decoded video quality was estimated with the help of packet delivery traces. The decodability of each frame was computed for simulcast and MVC encoding schemes according to packet traces. The successfully decoded frames were weighted with their original PSNR value whereas the undecodable frames received a 20dB penalty over their origi-

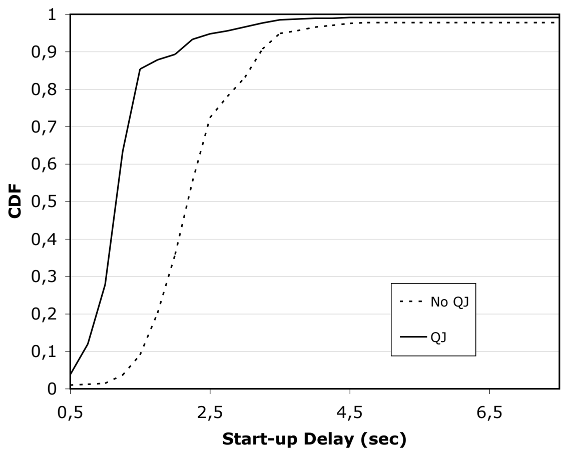

Fig. 4. Cumulative distribution of start-up delay with and without Quick Join (QJ)

nal quality to simulate frame freeze. The resulting curves can be seen in Fig. 3, where the solid black curves correspond to the original rate-distortion (R-D) performance of the multi-view video before streaming. Although the coding efficiency of MVC is much higher in this example, that advantage is almost lost at the receiver side due to the increased number of undecodable packets. Although, the PSNR performance of MVC is still about 1dB better in comparison to simulcast, it should also be noted that there are more frame freezes, which are very disturbing subjectively and therefore the quality difference between the encoded and delivered video quality is much larger for MVC.

The delay between sending the JOIN request and receiving the first data packet is called the start-up delay. In the normal join procedure the start-up delay is nearly equal to the delay of the six-way handshake: the data packets follow right after the ATTACH acknowledgement from the parent peer. As described in 2.1.3, we have implemented the Quick Join procedure. As it can be seen in Fig. 4, quick join considerably increases the probability of a shorter start-up delay, where the mean delay dropped from about 2 seconds to about 1 second with Quick Join.

4. CONCLUSIONS

We have presented a simple P2P streaming protocol and packet continuity results for different peer churn situations. A comparison of decoded video quality for simulcast and MVC encoded multi-view video was made using packet delivery traces. Although MVC has much better coding efficiency, due to packet losses caused by tree instabilities reduce that advantage significantly. Simulcast is relatively less affected by packet losses because the effects of lost packets are contained in one view. We have also introduced a quick join procedure which significantly cuts the start-up delay by temporarily allocating any spare capacity of an existing peer to the joining peer.

5. REFERENCES

[1] K. Mueller, P. Merkle, H. Schwarz, T. Hinz, A. Smolic, T. Oelbaum, and T. Wiegand, “Multi-view video coding based on H.264/AVC using hierarchical B-frames,” in *Picture Coding Symposium 2006*. PCS, 2006.

[2] A. Vetro, P. Pandit, H. Kimata, and A. Smolic, “Joint Draft 3.0 on Multiview Video Coding,” *Joint Video Team, Doc. JVT-W209*, 2007.

[3] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, “Client-driven selective streaming of multi-view video for interactive 3DTV,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1558 – 1565, November 2007.

[4] N. Özbek and M. Tekalp, “Scalable multi-view video coding for interactive 3DTV,” in *Proceedings of IEEE ICME 2006*. IEEE, 2006, pp. 213–216.

[5] M. Dröse, C. Clemens, and T. Sikora, “Extending Single-View Scalable Video Coding to Multi-View Based on H.264/AVC,” in *Proceedings of IEEE ICIP 2006*. IEEE, September 2006, pp. 2977–2980.

[6] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, “The feasibility of supporting large-scale live streaming applications with dynamic application endpoints,” *SIGCOMM 2004*, pp. 107–120, 2004.

[7] E. Setton and J. Apostolopoulos, “Towards quality of service for peer-to-peer video multicast,” *ICIP 2007*, vol. 5, pp. 81 – 84, Sept. 16 2007-Oct. 19 2007.

[8] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, “SplitStream: high-bandwidth multicast in cooperative environments,” *ACM 19th SOSP*, pp. 298–313, 2003.

[9] E. Setton, P. Baccichet, and B. Girod, “Peer-to-peer live multicast: A video perspective,” *Proceedings of the IEEE*, vol. 96, no. 1, pp. 25–38, Jan. 2008.

[10] E. Akyol, A. M. Tekalp, and M. R. Civanlar, “Adaptive Peer-To-Peer Video Streaming with Optimized Flexible Multiple Description Coding,” *ICIP 2006*, pp. 725–728, 2006.

[11] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, “Efficient Prediction Structures for Multiview Video Coding,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1461–1473, 2007.