DISTRIBUTED STREAMING VIA PACKET PARTITIONING

Jacob Chakareski and Pascal Frossard
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute - LTS4, CH-1015 Lausanne

ABSTRACT

We propose a system for adaptive streaming from multiple servers to a single receiver over separate network paths. Based on incoming packets, the receiver estimates the available bandwidth on every path and returns this information to the servers. An optimization algorithm is designed that enables the servers to independently partition the media packets among them according to the bandwidth information and such that the resulting video quality at the receiver is maximized. To this end, the algorithm takes advantage of a source pruning technique that preprocesses the media stream ahead of time. Simulation results demonstrate that the proposed streaming framework provides superior performance over a conventional transmission scheme that performs proportional packet scheduling based only on the available network bandwidth. Due to its low-complexity aspect, the framework is suitable for practical implementations of adaptive and efficient distributed streaming systems.

1. INTRODUCTION

Multiparty streaming has drawn considerable attention in recent years. One of the scenarios that fall into this category is distributed streaming, where multiple senders transmit packets over separate network paths to a single receiver, as illustrated in Figure 1.

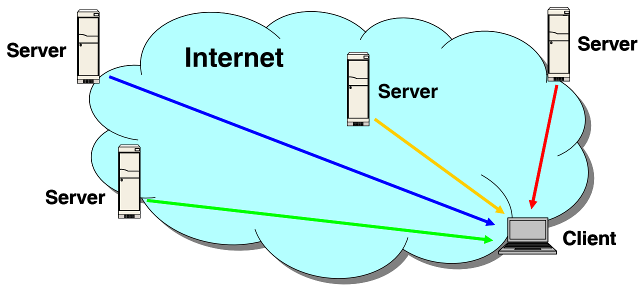

Fig. 1. Streaming from multiple senders to a single receiver.

To the best of our knowledge, the earliest work that studied the problem of transmission coordination among the multiple senders in distributed streaming is [1]. In this work, the authors proposed an algorithm that was run at the client and that performed rate allocation and packet partitioning among the senders. In a follow-up work, the authors combined the previously proposed algorithm with forward error correction for improved error resilience to burst packet loss [2]. Similarly, the works in [3, 4] considered receiver-driven control protocols that synchronized the senders' transmissions in a rate-distortion optimized way. For improved error-resilience, Multiple Description Coding (MDC) was employed at each sender to pre-encode (prior to transmission) a progressively encoded media content that was streamed afterwards to the client. Another related work is [5], where a rate-distortion optimization framework was proposed for packet scheduling in receiver-driven distributed video streaming. The paper established that the gains in performance due to server (path) diversity, relative to a single server (path) case, were dependent on the quality of the network paths in terms of packet loss and delay. Finally, the works in [6, 7] examined the performance of an MDC scheme for distributed video streaming in Content Delivery Networks (CDNs). The authors reported a 20-40% reduction in client video distortion, for the considered network conditions and topologies, relative to conventional CDNs that did not employ multiple description encoded video streams.

Differently from the prior work described above, in this paper we propose an optimization framework for coordinating the transmissions of the multiple senders in distributed sender-driven streaming. In particular, instead of computing the transmission schedules for each sender as in receiver-driven approaches, the client only monitors incoming packets on each network path to determine the available bandwidth in the forward directions of the paths. This information is then fed back to the senders to be used for computing appropriate transmission actions. An optimization algorithm is employed to partition the packets of the media stream independently, and yet in coordination, at every sender such that the resulting performance is maximized. To achieve this the algorithm relies on a source pruning technique that preprocesses the media stream ahead of time thereby providing for efficient and low-complexity distributed streaming.

The rest of the paper proceeds as follows. The specifics of the communication protocol employed between the senders and the client in the scenario under consideration are described in Section 2. Then, in Section 3 we briefly describe the source pruning technique that is used to preprocess the media packets. Section 4 describes next the proposed algorithm for distributed partitioning of the media packets across the multiple senders. Finally, in Section 5 we examine the performance of the proposed streaming system and compare it to that of a conventional distortion-agnostic system for distributed sender-driven streaming. The papers ends with concluding remarks provided in Section 6.

2. DISTRIBUTED SENDER-DRIVEN COMMUNICATION

Let there be M senders (servers) transmitting data units of a media presentation on M independent paths. The receiving client in turn monitors the forward-trip time of arriving packets and, based on these quantities, estimates the available bandwidth in the forward direction for each path. In particular, let $\tau_1^k, \ldots, \tau_P^k$ be the transmission delays experienced by the packets received by the client on path k in the last ΔT seconds. Then, the most recent estimate for the bandwidth (data rate) available on path k is computed by the client as $\widetilde{R}_k = (1/P)\sum_{j=1}^{P}(B_j/\tau_j^k)$. This is simply the average of the P most recent estimates of the available bandwidth on path k associated with the corresponding received packets, where B_j is the size of packet j in bits (or bytes). At the end of each estimation period ΔT the client returns to every sender a special acknowledgement packet[1] that contains the latest bandwidth estimates $\widetilde{R}_1, \ldots, \widetilde{R}_M$ for all paths. This information is then employed by the optimization algorithm from Section 4 to partition the media packets across the senders in the most efficient manner.

3. CLASSIFICATION OF PACKETIZED MEDIA

Recently, a source pruning technique for rate adaptation of packetized media has been proposed in [8]. The technique works as follows. Let $R_1, \ldots, R_K$ be a sequence of K monotonically increasing data rates. Packets of a media stream are classified into K sets $S_1, \ldots, S_K$, where the sets S_i are obtained by pruning (dropping packets from) the video source such the data rate of the pruned source does not exceed the corresponding rates R_i, for $i = 1, \ldots, K$. Note that a packet can be associated with multiple sets. While reducing the rate of the video source to R_i the algorithm selects to discard those data units from the packetized representation of the compressed source that will contribute to the smallest reduction in video quality. It is important to note that the pruning algorithm typically creates embedded subsets, i.e., for any two sets S_i and S_j such that $i < j$, it holds that $S_i \subset S_j$. For more details on the algorithm, the reader is referred to the cited reference [8].

4. STREAMING VIA PACKET PARTITIONING

We propose the following algorithm for partitioning the packets of a video stream among the M senders based on the available bandwidth. Each sender employs the technique from Section 3 to create a priori the subsets S_i for the video stream. A sufficiently large range of data rates is chosen so that it covers the possible bandwidth fluctuations encountered on the network paths. Moreover, a sufficiently large K is chosen so that there is fine (incremental) division of the data rate range $R_K - R_1$. Note that R_K should be chosen such that it corresponds to the encoding rate of the source, i.e., S_K contains all the packets from the video stream. Now, given the vector of estimated bandwidth values on each path $\widetilde{R}_1, \ldots, \widetilde{R}_M$ the packet partitioning algorithm proceeds as follows. If there is an $\mu \in \{1, \ldots, M\}$ such that it holds $\widetilde{R}_\mu \leq R_K$, then we are done. The video stream is simply streamed on any one of the paths for which the above is true[2]. Otherwise, for $m = 1, \ldots, M$ each sender m solves for the index $l(m)$ according to

$$l(m) = \arg\max_k R_k, \text{ s.t. } R_k \leq \sum_{i=1}^{m} \widetilde{R}_i, \tag{1}$$

and sends to the client the packets from the set $S_{l(m)} \setminus \bigcup_{i=1}^{m-1} S_{l(i)}$, where "$\setminus$" denotes the operator "set difference". In case there is an $m < M$ for which $l(m) = K$, there is no need to run the algorithm further, i.e., for the rest of the indices $m < j \leq M$. In other words, we would reach a point where we could send all the packets from the video stream on the first m paths. Note that employing such a procedure for constructing the partitions of packets sent on each path is possible because of the property that the sets S_i form embedded subsets, as mentioned earlier. We summarize the packet partitioning algorithm in Figure 2.

[1]Alternatively, these bandwidth estimates may be piggy-backed on regular acknowledgement packets.

[2]For example, the senders can agree ahead of time on the strategy where the stream is sent on the first (smallest) μ for which this condition holds.

```
Given $\widetilde{R}_1, \ldots, \widetilde{R}_M$,
  (0) If $\exists\, \mu \in \{1, \ldots, M\}$, s.t. $\widetilde{R}_\mu \leq R_K$
        Send set $S_K$ on path $\mu$.
        Exit.
  (1) Else
        Initialize: $m = 1, l(0) = 1$
        while $l(m-1) < K$ and $m \leq K$ do
            $l(m) = \arg\max_k R_k$, s.t. $R_k \leq \sum_{i=1}^{m} \widetilde{R}_i$
            [Index assignment]
            Send on path $m$: $S_{l(m)} \setminus \bigcup_{i=1}^{m-1} S_{l(i)}$
            [Packet partitioning]
            $m = m + 1$
  (2) End
```

Fig. 2. Distributed streaming via packet partitioning.

The algorithm can be generalized in a straightforward manner to the case when the networks paths exhibit packet loss in the forward direction, in addition to the varying bandwidth. In particular, let ϵ_m denote the erasure rate of packets sent to the client on path m, for $m = 1, \ldots, M$. These quantities can be estimated by the client based on missing sequence numbers of arriving packets and returned back to the senders together with the bandwidth estimates $\widetilde{R}_1, \ldots, \widetilde{R}_M$. The modification on the senders' part then consists of merely employing the effective bandwidth estimates $\widetilde{R}_1(1-\epsilon_1), \ldots, \widetilde{R}_M(1-\epsilon_M)$ in the packet partitioning algorithm.

5. SIMULATION RESULTS

This section examines the performance of the proposed framework for distributed streaming of packetized video content. The video content employed in the simulations are the test video sequences Foreman and Mother & Daugther in QCIF size encoded at 10 fps using an H.264 codec Each sequence is encoded with a constant quantization level at an average luminance (Y) PSNR of about 36 dB and a Group of Pictures (GOP) size of 20 frames, where each GOP consists of an I frame followed by 19 consecutive P frames. Performance is measured in terms of the average Y-PSNR of the reconstructed frames of a video content at the receiver. Frames that are not delivered on time are replaced by the receiver using previous frame error concealment.

There are $M = 2$ senders streaming video content to a client over two independent network paths. The network bandwidth in the forward direction of each path is randomly varying between a lower and an upper bound. In the simulations, random fluctuations of the available bandwidth occur every two seconds, while the time period ΔT employed at the client for bandwidth estimation is set to one second. The packet delay densities are assumed to be exponential functions and are inherently tied to the available bandwidth on the network paths. Finally, the play-out delay of the client application is set to one second.

In the first set of simulations, we examine adaptation to varying network bandwidth. In particular, the packet loss rates on the network paths are assumed to be zero, and we study how the proposed system performs in adjusting the streaming rate of the video content to the bandwidth variations of the underlying network. The range in which the network bandwidth is randomly varying, is given as $[BW_{min}, BW_{min} + 20]$ for one of the paths, and $[BW_{min} + 20, BW_{min} + 40]$ for the other network path, where BW_{min} is measured in kbps. Hence, the paths are asymmetric in terms of available bandwidth. In the simulations, we change BW_{min} across a certain range, and for each of its values we record the corresponding Y-PSNR performance of the proposed system, henceforth denoted *PackPart*. For comparison purposes, in the simulations we also examine the performance of a conventional system, denoted *Baseline*, that performs proportional packet scheduling based only on the available network bandwidth. In particular, the two senders split the packets of the video stream in proportion to the bandwidth estimates provided by the client. Packet dropping decisions in *Baseline* are performed randomly without taking into account their specific distortion importance. In other words, *Baseline* is distortion-agnostic.

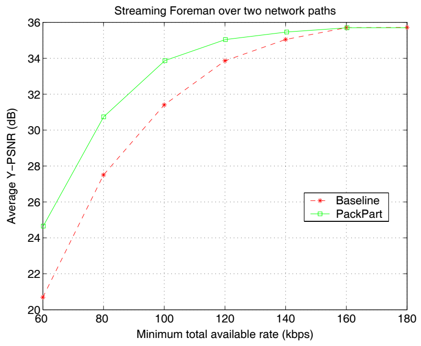

Fig. 3. Bandwidth adaptation of Foreman.

In Figure 3 we show the performances of the two systems for streaming Foreman as a function of the minimum transmission bandwidth available on both paths during a session, i.e., $2 * BW_{min} + 20$. It can be seen that *PackPart* provides an improved performance over the baseline system almost over the whole range of values considered for the minimum overall bandwidth. The gains in performance are especially significant in the lower half of the bandwidth range. For example, at 60 kbps minimum overall bandwidth there is a difference of 4 dB in performance between the two systems. The improved performance of *PackPart* is due to the fact that the packet partitions from which the senders stream the video data are selected based on the subsets of packets S_i. These in turn are selected such that they correspond to the maximum possible video quality for the corresponding available data rates, as explained in Section 3. On the other hand, *Baseline* performs bandwidth adaptation without treating the various packets preferentially, as it is distortion-agnostic. Therefore, some more important packets may be dropped at the expense of others, less important ones, which would ultimately lead to a degradation in video quality at the client.

Finally, it can be seen from Figure 3 that both systems perform alike for a sufficiently large minimum total bandwidth. This is expected as here there is sufficient bandwidth available throughout the session to ensure timely delivery of all packets to the receiver in the case of each system. In other words, no packets need to be dropped anymore due to a mismatch between the network bandwidth variations and the dynamically varying source encoding rate.

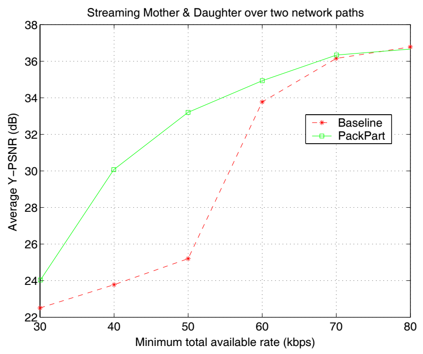

Fig. 4. Bandwidth adaptation of Mother & Daughter.

Similar relative performances for the two systems are observed for streaming Mother & Daughter, as shown in Figure 4. It can be seen that again *PackPart* outperforms *Baseline* almost over the whole range of bandwidth values under consideration, with gains reaching up to 6-8 dB in the lower half of the bandwidth range. However, when the overall available bandwidth reaches a value at which no bandwidth adaptation is needed, *PackPart* and *Baseline* perform identically, as illustrated by their performances for 80 kbps minimum overall bandwidth.

Next, we examine the performances of *PackPart* and *Baseline* when adapting to both bandwidth variations and random packet loss. In particular, in addition to dynamic bandwidth variations the communication channels also exhibit random packet erasures now. Therefore, packets will be lost during transmission and a streaming system needs to decide then whether it would retransmit (old) previously transmitted packets or it would send new packets that have not been sent before. This trade-off is necessitated by the fact that the available bandwidth is insufficient to support sending all the packets together, including the retransmissions. As we did previously, in the simulations we measure the performances of *PackPart* and *Baseline* as a function of the available data rate, but now in the presence of packet loss. Specifically, we examine packet loss rates (ϵ_F) of 5% and 10% on the forward channels from the senders to the receiver.

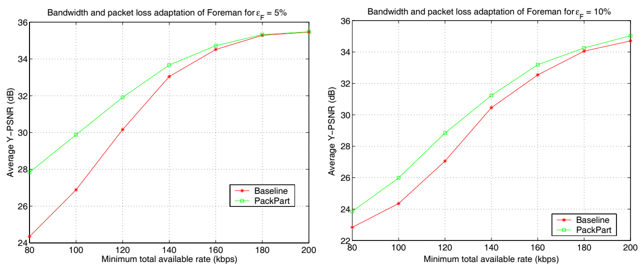

Fig. 5. Bandwidth and packet loss adaptation of Foreman: (left) $\epsilon_F = 5\%$ and (right) $\epsilon_F = 10\%$.

In Figure 5, we show the performances of the two systems for streaming Foreman in this scenario. Firstly, it can be seen that both of them exhibit a degraded performance relative to the corresponding results shown in Figure 3, where only bandwidth adaptation is performed. This is expected as now in addition to discarding packets due to bandwidth variations, each system also has to take into account retransmissions of lost packets. Therefore, higher data rates on the communication channels are needed to achieve the same Y-PSNR performance relative to the case of bandwidth adaptation only, as seen from Figure 3. Secondly, it can be also seen that the performances of *PackPart* and *Baseline* degrade with increasing the packet loss rate. For example, for $\epsilon_F = 5\%$, both systems exhibit a Y-PSNR performance within the 33-34 dB range when the total available data rate on the channels is 140 kbps. However, that performance reduces to being in the range of 30-32 dB for the same data rate at packet loss rate of 10%. Such a performance behaviour across the two systems is also expected, as the increasing loss rate reduces the number of packets that can be delivered on time to the receiver, given the selected play-out delay.

Finally, it should be pointed out that *PackPart* provides the most significant gains over *Baseline* in the lower end of data rates under consideration, especially for $\epsilon_F = 5\%$, as seen from Figure 5 (left). As explained earlier in the context of bandwidth adaptation, this is due to the fact that *PackPart* can better trade-off the importance of each packet for the available data rate on the channels relative to the conventional system that is distortion agnostic. However, as the packet loss rate is increased *PackPart* cannot take so much advantage of the knowledge of the packets' distortion importance, since even though packets are prioritized in terms of transmission they are more likely to be lost now, and there is not enough data rate to perform loss recovery by retransmission. On the other hand, when there is sufficient data rate available on the channels, both systems can deal effectively with packet loss by performing retransmissions, as their similar performances in the upper end of the data rate range illustrate.

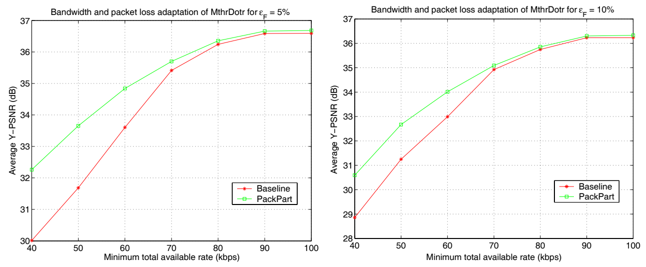

Fig. 6. Bandwidth and packet loss adaptation of Mother & Daughter: (left) $\epsilon_F = 5\%$ and (right) $\epsilon_F = 10\%$.

In Figure 6 we show the corresponding performances of *PackPart* and *Baseline* for streaming Mother & Daughter. It can be seen that the relative performances between the two systems across the different data rates and packet loss rates under consideration exhibit a similar behaviour to those for the case of Foreman. In particular, the optimized system outperforms more significantly the conventional system in the lower end of data rates and especially for lower packet loss rates. As the packet loss rate increases the performances of *PackPart* and *Baseline* degrade significantly and therefore become more alike. In the same spirit, as the available data rate is sufficiently increased, the two systems perform alike again, but now due to the fact that there is enough bandwidth to recover (almost completely) from packet losses by retransmissions, as shown in Figure 6.

6. CONCLUSIONS

We have presented a system for adaptive distributed video streaming in sender-driven way. The system consists of an optimization algorithm for partitioning the video packets across the individual senders, combined with a bandwidth estimation technique that is performed at the receiving client. Using the algorithm and the bandwidth estimates provided by the client, the senders can independently, but still in coordination, decide what the most important packets are to transmit on each path such that the overall performance is maximized, for the given network bandwidth. This is enabled by employing a priori classification of the media packets provided by pruning the compressed video stream at different data rates. In the simulation of the proposed system, it is established that significant performance gains are observed over a conventional distortion-agnostic system for distributed streaming. This is true as long as bandwidth adaptation of the video content needs to be performed due to the disparity between the available network bandwidth and the encoding rate of the video content, both of which are varying over time. However, in the presence of substantial packet loss the advantage of knowing the distortion importance of the video packets when scheduling the transmissions reduces due to the increased likelihood of uncertain delivery to the receiver caused by losses during transmission. It should be noted that the proposed system provides significant gains over the conventional technique while maintaining comparable online complexity, when data unit characteristics have been pre-computed off-line. Finally, even if only two servers have been used in the simulation, we do not expect the results to be significantly different for a larger number of servers. As it is observed in multipath streaming, the benefit of distributed streaming may however saturate for a small number of servers, typically 3-5.

7. REFERENCES

[1] T. Nguyen and A. Zakhor, "Distributed video streaming over internet," in *Proc. Multimedia Computing and Networking*, San Jose, CA, Jan. 2002, SPIE, vol. 4673, pp. 186–195.

[2] T. Nguyen and A. Zakhor, "Distributed video streaming with forward error correction," in *Proc. Int'l Packet Video Workshop*, Pittsburg, PA, Apr. 2002.

[3] A. Majumdar, R. Puri, and K. Ramchandran, "Distributed multimedia transmission from multiple servers," in *Proc. ICIP*, Rochester, NY, USA, Sept. 2002, IEEE.

[4] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Network-adaptive video streaming using multiple description coding and path diversity," in *Proc. ICME*, Baltimore, MD, USA, July 2003, IEEE.

[5] J. Chakareski and B. Girod, "Server diversity in rate-distortion optimized streaming of multimedia," in *Proc. ICIP*, Barcelona, Spain, Sept. 2003, IEEE.

[6] J. Apostolopoulos, T. Wong, W.-T. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proc. Infocom*, New York City, NY, USA, June 2002, IEEE.

[7] J. Apostolopoulos, W.-T. Tan, and S. Wee, "Performance of a multiple description streaming media content delivery network," in *Proc. ICIP*, Rochester, NY, USA, Sept. 2002, IEEE.

[8] J. Chakareski and P. Frossard, "Low-complexity adaptive streaming via optimized a priori media pruning," in *Proc. Workshop on MMSP*, Shanghai, China, Oct./Nov. 2005, IEEE.