

MULTI-STREAM PARTITIONING AND PARITY RATE ALLOCATION FOR SCALABLE IPTV DELIVERY

Jacob Chakareski and Pascal Frossard

Signal Processing Laboratory - LTS4, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

ABSTRACT

We address the joint problem of clustering heterogeneous clients and allocating scalable video source rate and FEC redundancy in IPTV systems. We propose a streaming solution that delivers varying portions of the scalably encoded content to different client subsets, together with suitably selected parity data. We formulate an optimization problem where the receivers are clustered depending on the quality of their connection so that the average video quality in the IPTV system is maximized. Then we propose a novel algorithm for determining optimally the client clusters, the source and parity rate allocation to each cluster, and the set of serving rates at which the source+parity data is delivered to the clients. We implement our system through a novel design based on scalable video coding that allows for much more efficient network utilization relative to the case of source versioning. Through simulations we demonstrate that the proposed solution substantially outperforms baseline IPTV schemes that multicast the same source and FEC streams to the whole client population, as is commonly done in practice today.

1. INTRODUCTION

Internet Protocol Television (IPTV) is an emerging multimedia application that is expected to gain a significant share of the broadcast media market. IPTV systems deliver conventional TV content via IP multicast over privately owned and managed broadband networks. Customers access the service from a local switching office using their last mile Internet connection. In order to combat packet loss experienced during the streaming of the content over the access network of the customers the provider includes forward error correction (FEC) packets in the multicast distribution.

The rate of the FEC packets in an IPTV distribution is typically fixed and determined ahead of time. It usually corresponds to a conservative estimate of the loss rate that the media packets could experience during streaming, as assessed by the IPTV provider. However, despite its appeal due to its simplicity this approach in general reduces the efficiency of the content delivery process. In particular, customers whose access networks exhibit loss rates below the protection level of the FEC packets will be unnecessarily penalized with a drop in video quality. On the other hand, customers experiencing packet loss rates exceeding the correction capability of the FEC data will observe a significant degradation in video quality.

The prospects of the above scenario are quite real at present as the receiving clients of an IPTV multicast typically exhibit a variety of packet loss rates. Today, we employ a plethora of broadband technologies for our Internet access links starting from DOCSIS (cable) or DSL, to Ethernet or FTTx (fiber), and to wireless (Wi-Fi, WiMAX, and cellular). Each of them is characterized with a markedly distinct profile in terms of bandwidth, packet loss, and

delay. The IPTV clients' heterogeneity is further amplified by the fact that customers frequently rebroadcast the content to different TV sets in their homes again using a range of different networking technologies.

In this paper, we consider clustering the IPTV clients according to their packet loss and bandwidth characteristics. Varying amounts of source and parity data are then delivered at different overall serving rates to the individual client clusters in order to optimize the overall performance of the system. Efficient network utilization is achieved via a novel implementation of the system based on scalable video coding. We formulate an optimization problem whose objective is to determine the client clusters and the amount of source and parity data served to each cluster such that the overall video quality that the system delivers is maximized. We propose a novel iterative algorithm for solving this resource allocation problem and we show that the algorithm clearly outperforms common solutions where the rate allocation is uniform across the whole client population.

Client partitioning in multicast streaming systems has been studied in [1] where a dynamic programming algorithm is proposed for computing the rate allocation that maximizes the sum of the receivers' utilities. The authors in [2] address the problem of multicast server selection for adapting to the dynamics of streaming networks. The design of multicast solutions based on layered video streams has also been discussed in [3] and a comparison of delivery solutions based on versions or layers is provided in [4]. The case of lossy scenarios is studied in [5], where clients are partitioned into two classes in order to maximize the decoding quality of a layered multiple description coding scheme. Finally, in [6] the authors design a system for IPTV multicast over wireless LANs where a proxy server adapts the amount of FEC protection sent on the wireless link based on client feedback.

2. SYSTEM DESCRIPTION

In our system, the video content is scalably encoded and broadcasted on the backbone network of the IPTV provider. There is a set of FEC streams that are broadcasted together with the source data. They provide different levels of packet loss protection to the IPTV clients. Then, a client subscribes to a portion of the scalable stream and its corresponding protection level, as computed by our optimization algorithm that clusters the clients according to their bandwidth and packet loss characteristics. To control system complexity, there is a limited set of serving rates at which source and parity data can be delivered to the clients.

Consider for instance the illustration in Figure 1 where the source and parity rates assigned to four clients are shown. Clients C_1 and C_2 are low bandwidth clients, while C_3 and C_4 are high bandwidth clients. As C_1 and C_3 exhibit higher packet loss characteristics than C_2 and C_4 , respectively, the optimization assigns to the former correspondingly lower source rates, but higher FEC rates, i.e., $r_s^1 < r_s^2$, $r_s^3 < r_s^4$, $r_p^1 > r_p^2$, and $r_p^3 > r_p^4$. The over-

This work was supported by the Swiss National Science Foundation under Ambizione grant PZ00P2-126416.

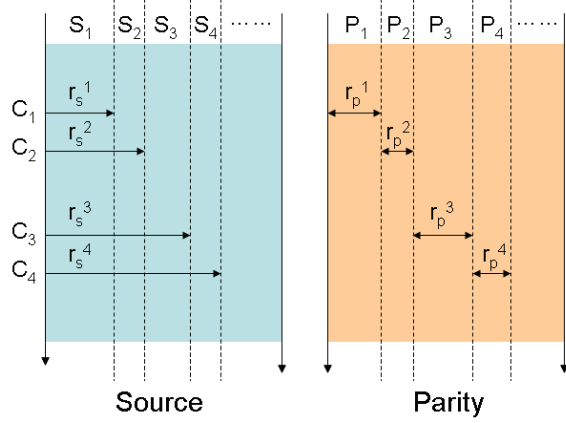


Fig. 1. Source and parity streams.

all system bandwidth at which data is delivered to client clusters featuring similar bandwidth characteristics is kept constant, e.g., $r_s^1 + r_p^1 = r_s^2 + r_p^2 = B_{\text{low}}$. Individual clients receive the allocated data (source + parity) by subscribing to the corresponding multi-cast streams, e.g., client C_1 subscribes to streams S_1 and P_1 , while client C_3 subscribes to streams S_1, S_2, S_3 , and P_3 , as illustrated in Figure 1.

3. RESOURCE ALLOCATION PROBLEM

Let B_l denote one of the L serving rates at which the system delivers source and parity data to its customers, as introduced earlier. For a given B_l , for $l = 1, \dots, L$, we are interested in clustering the client population into M_l distinct groups such that the overall video quality is maximized. In particular, each client cluster $i = 1, \dots, M_l$ will be served content at overall bandwidth B_l but different media rates r_i . This in turn corresponds to employing a different FEC code (k_i, n) for the IPTV content served to each cluster i , where k_i represents the number of source (media) packets and n denotes the block of the code in packets. In the rest of the exposition, we will drop the index l on all variables, save for B_l , to ease the notation.

Now, let $Q(r_i) = Q(k_i)$ denote the video quality of the content when its data rate is $r_i^s = (k_i/n)B_l$. Furthermore, let $[e_{i-1}, e_i]$ denote the range of loss rates associated with the clients in cluster i . Then, the cumulative expected video quality for the clients in this packet loss rate range can be computed as

$$\int_{e_{i-1}}^{e_i} E[Q(k_i), e] f(e) de, \quad (1)$$

where $E[Q(k_i), e]$ denotes the expected video quality for the IPTV content when served in the presence of packet loss rate e and protected with an FEC code (k_i, n) . Furthermore, $f(e)$ denotes the number of IPTV clients experiencing packet loss rate e on their access links. Assuming iid loss, $E[Q(k_i), e]$ can be computed as

$$E[Q(k_i), e] = Q(k_i) \sum_{j=k_i}^n \binom{n}{j} (1-e)^j e^{n-j}. \quad (2)$$

Finally, using (1) we can characterize the expected video quality $E[Q|B_l]$ for the entire client population served at rate B_l as

$$E[Q|B_l] = \frac{1}{\int_{e_0}^{e_M} f(e) de} \sum_{i=1}^M \int_{e_{i-1}}^{e_i} E[Q(k_i), e] f(e) de. \quad (3)$$

As stated earlier, we are interested in optimizing $E[Q] = \sum_l E[Q|B_l]$ over the configurable parameters of the IPTV system. In particular, let $\mathbf{k} = (k_1, \dots, k_M)$ denote the vector of source rates for the client clusters, let $\mathbf{e} = (e_0, \dots, e_M)$ denote the boundaries of the client groups as assigned to individual clusters, and lastly let M denote the number of sessions. Furthermore, let $\mathbf{B} = (B_1, \dots, B_L)$ denote the vector of serving rates at which the system delivers the FEC encoded content to the clients. Finally, the optimization problem of interest can be written as

$$\max_{\mathbf{k}, \mathbf{e}, M, \mathbf{B}, L} E[Q], \quad (4)$$

where the constraints of the optimization on the vectors \mathbf{e} and \mathbf{k} are explained in the next section.

4. OPTIMIZATION ALGORITHM

Here, we design an iterative coordinate descent algorithm for solving (4). In particular, each of the variables to be optimized, i.e., $\mathbf{k}, \mathbf{e}, M, \mathbf{B}, L$, can be considered as one coordinate dimension of the optimization problem in (4). Hence, we propose to optimize over one coordinate while keeping the others fixed, in an iterative fashion, until convergence. Finally, without lack of generality the first and last edges of the client population's binning can be selected as $e_0 = 0$ and $e_M = 1$.

Now, for a given B_l , for $l = 1, \dots, L$, let $k_i^{(0)} \in \{k_{\min}, \dots, n\}$, for $i = 1, \dots, M$, denote the initial values of the source rate at which the content is served to each client cluster. Here, k_{\min} denotes the minimum acceptable source rate below which the content cannot be delivered without resulting into customer dissatisfaction. Similarly, let $e_1^{(0)} < e_2^{(0)} < \dots < e_{M-1}^{(0)}$ denote the initial loss rate values employed for assigning the clients to different clusters. Analogously, let $B_1^{(0)} < B_2^{(0)} < \dots < B_L^{(0)}$ denote the initial serving rate values employed for delivering the content to the client population. Finally, the initial number of sessions M and rate values L are selected to be sufficiently large.

Then, at every iteration $j = 1, 2, \dots$ we run three loops consecutively. In the first one, we adjust the packet loss rate bins as follows. For $i = 1, \dots, M-1$ we solve

$$e_i^{(j)} = \arg \max_{e_i \in (e_{i-1}^{(j-1)}, e_{i+1}^{(j-1)})} \left(\int_{e_{i-1}^{(j-1)}}^{e_i} E[Q(k_i^{(j-1)}), e] f(e) de + \int_{e_i}^{e_{i+1}^{(j-1)}} E[Q(k_{i+1}^{(j-1)}), e] f(e) de \right). \quad (5)$$

Subsequently, we recompute the media data rates in the second loop. In particular, for $i = 1, \dots, M$ we solve

$$k_i^{(j)} = \arg \max_{k_i \in \{k_{\min}, \dots, n\}} \int_{e_{i-1}^{(j)}}^{e_i^{(j)}} E[Q(k_i), e] f(e) de. \quad (6)$$

Finally, in the third loop we adjust the system bandwidth values B_l . Specifically, for $l = 1, \dots, L$ we solve

$$B_l^{(j)} = \arg \max_{B_l \in (B_{l-1}^{(j-1)}, B_{l+1}^{(j-1)})} E[Q|B_l]. \quad (7)$$

Note that in (7) we choose $B_0 = B_{\min}$ and $B_{L+1} = B_{\max}$. These represent respectively the minimum and maximum rate values that can be employed for serving the clients.

Next, we check whether the recomputed source rates satisfy $k_i^{(j)} \neq k_{i+1}^{(j)}$, for $i = 1, \dots, M - 1$. If this condition is satisfied, then the algorithm proceeds. Otherwise, the number of different clusters in the system being served at a given B_l needs to be reduced first. For instance, let $k_i^{(j)} = k_{i+1}^{(j)}$ for one specific $i \in \{1, \dots, M\}$. Then, clusters i and $i + 1$ are being served at the same source rate. Therefore, we merge these two segments together and assign all the corresponding clients to cluster i only. Cluster $i + 1$ is removed from the system as unnecessary and we update $M = M - 1$. Correspondingly, we adjust the vector of source rates $\mathbf{k}^{(j)} = (k_1^{(j)}, \dots, k_i^{(j)}, k_{i+2}^{(j)}, \dots, k_M^{(j)})$ and the vector of loss rate values employed for the client binning $\mathbf{e}^{(j)} = (e_1^{(j)}, \dots, e_i^{(j)}, e_{i+2}^{(j)}, \dots, e_M^{(j)})$. We similarly remove all (if any) duplicate values in the vector $\mathbf{B}^{(j)}$.

Finally, at the end of iteration j we compute the expected video quality $E[Q]^{(j)}$ using (3). If $E[Q]^{(j)} = E[Q]^{(j-1)}$ we have converged and the algorithm exits. Otherwise, the optimization proceeds with iteration $j + 1$. In Figure 2 below, we provide a formal algorithmic description of the optimization.

```

(0) Initialize  $\mathbf{k}^{(0)}$ ,  $\mathbf{e}^{(0)}$ ,  $M$ ,  $L$ ,  $\mathbf{B}$ ,  $E[Q]^{(0)} = 0$ 
(1) for  $j = 1, 2, \dots$  do
    Compute  $\mathbf{e}^{(j)}$  using (5)
    Compute  $\mathbf{k}^{(j)}$  using (6)
    Compute  $\mathbf{B}^{(j)}$  using (7)
    If  $\exists i$  such that  $k_i^{(j)} = k_{i+1}^{(j)}$ 
        Update  $M$ ,  $\mathbf{k}^{(j)}$ ,  $\mathbf{e}^{(j)}$ 
    If  $\exists l$  such that  $B_l^{(j)} = B_{l+1}^{(j)}$ 
        Update  $L$ ,  $\mathbf{B}^{(j)}$ 
    Compute  $E[Q]^{(j)}$  using (3)
    If  $E[Q]^{(j)} = E[Q]^{(j-1)}$ 
        BREAK

```

Fig. 2. Proposed optimization algorithm.

Convergence of the proposed algorithm is guaranteed as the objective function is bounded from above and at every subsequent iteration of the algorithm it is monotonically increasing (or at least non-decreasing). We have observed in our numerical experiments that the optimization rapidly converges, typically within a very small number of iterations. Unfortunately, a global convergence of gradient ascent (or descent) algorithms of this type is very difficult to prove because of the complexity of their objective functions. Lastly, it should be mentioned that in our experiments we usually initialize the client bins $(e_1^{(0)}, e_2^{(0)}, \dots)$ to linearly span the possible range of packet loss rates that the client population exhibits. We similarly select the initial source rates $(k_1^{(0)}, k_2^{(0)}, \dots)$ for the various client segments at the onset of the optimization.

5. SIMULATION RESULTS

Here, we examine the performance of the proposed optimization algorithm for IPTV multicast to heterogeneous clients. The video content of interest comprises the CIF sequence Foreman encoded at 30 fps and multiple bitrates using the scalable extension (SVC) of an

H.264 codec. Then, depending on the specific source rate computed by the optimization, a multicast server delivers one of these encodings to its target audience. The minimum and maximum system bandwidth values B_{\min} and B_{\max} that were introduced earlier were set to 500Kbps and 2.2Mbps, respectively. The FEC block size that we employed in our experiments is 64 packets. The maximum allowed rate for the corresponding FEC packets was set to 20% of the overall serving rate B_l .

For our numerical experiments, we synthesized a client population via an exponential distribution function $f(e)$, as introduced earlier. There are in total more than 200K clients in the population. The minimum packet loss rate encountered among the clients is 0.5% while the maximum is 20%. The optimization from Section 4 selected $L = 2$ to be the optimal number of serving rates for the system. Furthermore, the optimal B_1 and B_2 were computed to be 1200Kbps and 2000Kbps, respectively. In Figure 3, we show the function $f(e)$ together with the computed source rates for the client clusters in the case of B_1 . The vertical lines in Figure 3 delineate the different client segments according to their packet loss rate, as described in Section 4.

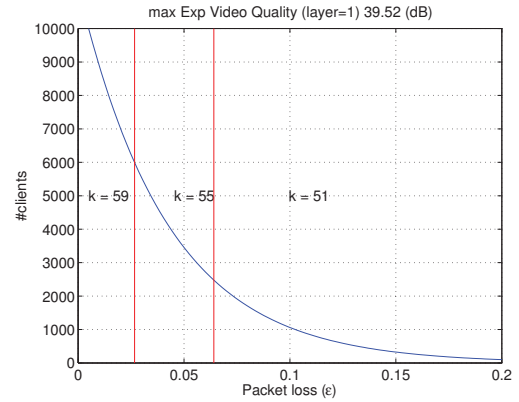


Fig. 3. IPTV multicast via three sessions for B_1 : Client bins (marked with red vertical lines); Corresponding source rates k_i (bold text); Client distribution function $f(e)$ (blue solid line).

We can see from Figure 3 that the number of required clusters that the algorithm computed is three. The corresponding source rates are $k_1 = 59$, $k_2 = 55$, and $k_3 = 51$. As expected, the optimization progressively reduces the source rate assigned to the client clusters as their packet loss rate increases. Specifically, in order to compensate for the increasing packet loss the algorithm decides to gradually increase the FEC packets' rate for the associated clusters thereby maintaining video quality at the maximum achievable level. Lastly, the expected video quality (Y-PSNR) for the client population at B_1 , as computed by our algorithm, is 39.52 dB.

Next, in Figure 4 we show the corresponding distribution of source rates in the case of B_2 . Note that here due to the higher serving rate the algorithm actually computes four to be the optimal number of client clusters for the given packet loss distribution. In particular, the corresponding source rates are $k_1 = 60$, $k_2 = 56$, $k_3 = 53$, and $k_4 = 51$. As in the case of B_1 , also here the optimization progressively reduces the source rate assigned to a cluster as the corresponding packet loss rate increases. The expected video quality for the client population served at B_2 is normally higher (41.76dB).

Next, we compare the multicast performance of our algorithm against those of two conventional schemes for delivering multi-

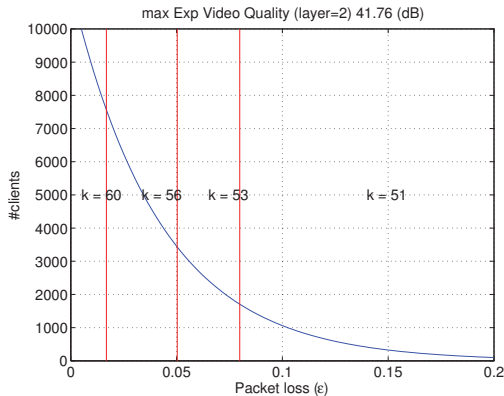


Fig. 4. IPTV multicast via four sessions for B_2 : Client bins (marked with red vertical lines); Corresponding source rates k_i (bold text); Client distribution function $f(\epsilon)$ (blue solid line).

dia content in an IPTV context. Both of these techniques employ a single error protection level for the entire client population. The only difference between the two is the rate of their FEC packets that they employ to this end. The first scheme, denoted henceforth *UnifMean*, selects the rate of its parity packets to correspond to the middle point of the range of packet loss rates exhibited by the client population. The second technique, denoted henceforth *UnifCentr*, selects the rate of its parity packets to correspond to the weighted (by $f(\epsilon)$) average of all possible packet loss rates. In other words, the rate of the FEC packets in this case corresponds to the centroid of the distribution of loss rates weighted by the distribution of number of clients as a function of packet loss, i.e., $\int \epsilon f(\epsilon) d\epsilon / \int f(\epsilon) d\epsilon$.

In Figure 5 below, we show a bar graph representing the video quality performance (Y-PSNR) of all three schemes for each of the three client clusters in the case of B_1 . In addition, we overlay in Figure 5 horizontal lines representing the average performance (in dB) of each scheme across the whole client population. First, it should be noted that our optimization algorithm, denoted henceforth *Opt*, performs identically to *UnifMean* in the case of the first client cluster, while it outperforms for 0.8dB *UnifMean* on the second client cluster. The performance gap between the two techniques, at the expense of *UnifMean*, then increases further to 8.5 dB in the case of the third client segment, as shown in Figure 5.

Furthermore, the comparison between *UnifMean* and *UnifCentr* is even more interesting. By selecting the rate of its FEC packets to correspond roughly to the mid range of observed packet loss rates *UnifMean* is able to perform practically as well as *Opt* on the first cluster, while moderately under performing on the second one. Contrarily, *UnifCentr* chooses its FEC packet rate to correspond to the far left of the packet loss rate axis, as necessitated by the distribution $f(\epsilon)$. This causes *UnifCentr* to non-trivially (1.8 dB) under perform the other two schemes even on the first client cluster, as the rate of its FEC packets is quite small and appropriate only for the clients exhibiting very small packet loss rates. The consequences of the imbalance between source and parity rates in the case of *UnifCentr* are accentuated even further when one moves on to the subsequent client clusters. This is illustrated well in Figure 5 by the continuously increasing degradation in video quality for *UnifCentr* over client cluster indices two and three. For instance, for the second client segment the performance of *UnifCentr* has already dropped for 11.3 dB relative to its own performance on the first cluster. Similarly, the gap

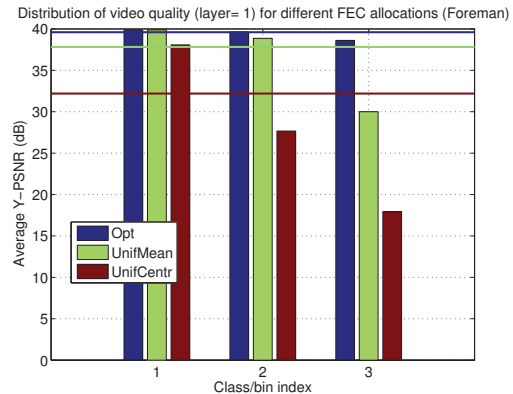


Fig. 5. CIF Foreman (B_1): Y-PSNR (dB) per client segment and overall for three multicast schemes: *Opt* (dark blue), *UnifMean* (green), and *UnifCentr* (brown).

in performance between *Opt* and *UnifMean* on one hand and *UnifCentr* on the other has also substantially increased between the two client segments. We observed similar results in regard to the performances of the three multicast schemes in the case of B_2 . These are not included here due to space constraints.

6. CONCLUSIONS

Current IPTV systems are often designed in a conservative manner and deliver the same multicast stream to heterogeneous clients that are subscribed to the same TV channel. Instead, we propose a novel resource allocation algorithm that clusters the clients according to their heterogeneous characteristics and then delivers varying amounts of source and parity data to each cluster such that the overall performance of the system is maximized. To reduce the bandwidth requirements of the IPTV delivery we employ scalably encoded content that allows us to efficiently stream the data to the clients via a number of simultaneous multicast sessions. Our strategy significantly outperforms baseline solutions that only deliver a single stream to all clients, even when the corresponding source and channel rates are optimized for a maximum average quality over the whole population of receivers.

7. REFERENCES

- [1] Y. Yang, M. Kim, and S. Lam, "Optimal partitioning of multicast receivers," in *Proc. IEEE ICNP*, Nov 2000, pp. 129–140.
- [2] Z. Fei, M. Ammar, and E. Zegura, "Multicast server selection: problems, complexity, and solutions," *IEEE JSAC*, Jan 2002.
- [3] S. McCanne et al., "Low-complexity video coding for receiver-driven layered multicast," *IEEE JSAC*, Aug 1997.
- [4] T. Kim and M. Ammar, "A comparison of layering and stream replication video multicast schemes," in *Proc. NOSSDAV*, 2001.
- [5] P. A. Chou, H. J. Wang, and V. N. Padmanabhan, "Layered multiple description coding," in *Proc. Packet Video*, Apr. 2003.
- [6] M. Wu, S. Makharia, H. Liu, D. Li, and S. Mathur, "IPTV multicast over wireless lan using merged hybrid ARQ with staggered adaptive FEC adaptive systems for improved media streaming experience," *IEEE Trans. Broadcasting*, June 2009.