Forward Error Correction for Multipath Media Streaming

Dan Jurca, *Member, IEEE*, Pascal Frossard, *Senior Member, IEEE*, and Aleksandar Jovanovic

***Abstract*— We address the problem of joint optimal rate allocation and scheduling between media source rate and error protection rate in scalable streaming applications over lossy multipath networks. Starting from a distortion representation of the received media information at the client, we propose a novel optimization framework in which we analyze the performance of the most relevant forward error correction and scheduling techniques. We describe both optimal and heuristic algorithms that find solutions to the rate allocation and scheduling problem, and emphasize the main characteristics of the compared techniques. Our results show that efficient unequal error protection schemes improve the quality of the streaming process. At the same time we emphasize the importance of priority scheduling of the information over the best available network paths, which outperforms traditional first-in-first-out models or network flooding mechanisms.**

***Index Terms*— Forward error correction, multipath networks, video streaming.**

I. Introduction

MEDIA STREAMING over the Internet remains challenging, mainly due to the best effort packet transport medium. Efficient and adaptive streaming systems need to be derived in order to bridge the gap between the stringent requirements of the media application and the unreliable conditions of the network infrastructure. Lately, multipath video streaming has emerged as a viable solution to some of the transport issues. The benefits of multipath routing in media streaming systems are presented in [1] and [2]. Among the main benefits of using multiple paths between a media server and a client we enumerate: 1) the reduction in correlation between packet losses; 2) increased throughput; and 3) ability to adjust to variations of congestion patterns on different parts of the network. Moreover, error robustness mechanisms can be employed on top of the routing techniques, in order to preserve the transmission quality over lossy network environments [3]–[5]. For real-time multimedia applications, or streaming sessions where the client playback delay is small, proactive strategies for error robustness are advisable, as they are much faster compared to traditional Automatic Repeat Request-based techniques. Forward error correction (FEC) is the main technique to provide a more reliable packet transmission in erasure networks. FEC usually provides additional redundant packets, which are sent along the data packets to the client. As long as the client receives enough data and redundant packets, it is able to reconstruct all original data packets.

However, the problem of application-specific error protection in multipath streaming systems has been given little attention so far. In lossy network scenarios, where media packets are prone to transmission erasures, it is important to chose the right amount of redundancy and the proper distribution between the source and channel rate in order to guarantee successful decoding at the end client. Moreover, given the multipath network scenario, additional choices concerning the efficient scheduling of media and redundant information over the multiple transmission paths should be made. In this paper we address the problem of joint optimal rate allocation and scheduling between media source rate and error protection rate in lossy multipath networks.

Based on a generic distortion model for layered encoding video streams, which takes into account possible packet transmission losses, we formulate a general optimization problem that targets an optimal balance between video source rate and forward error correction rate, given a constraint on total network resources. The optimal solution for our problem differs with the choice of FEC strategies and scheduling schemes. Hence, based on the most common FEC and scheduling techniques, we propose several concrete instances of this problem and we compute the optimal solutions. In particular, we address the equal and unequal FEC schemes, along with prioritized or unprioritized scheduling techniques for layered video coding over multiple network paths. At the same time, we offer fast heuristic algorithms that provide good results for our problem with small computational effort. Our results confirm that it is always best to stream on the best network paths first, and that fully utilizing the network resources is not always optimal in terms of average media quality. At the same time, we show the benefits of unequal error protection (UEP), and we identify the tradeoff between rate allocation optimality and service granularity in real systems.

Manuscript received October 25, 2007; revised August 7, 2008 and December 23, 2008. First version published May 12, 2009; current version published September 16, 2009. This work has been funded by the Swiss NSF, under Grant PP002-68737. This paper was recommended by Associate Editor J. Cai.

D. Jurca was with the Ecole Polytechnique Féderale de Lausanne (EPFL), Lausanne, Switzerland. He is currently with the DOCOMO Communications Laboratories GmbH, Munich, Germany (e-mail: jurca@docomolab-euro.com).

P. Frossard is with the Ecole Polytechnique Féderale de Lausanne (EPFL), Signal Processing Laboratory, Lausanne, Switzerland (e-mail: pascal.frossard@epfl.ch).

A. Jovanovic is with the Ecole Polytechnique Federale de Lausanne, The Electronics and Signal Processing Laboratory, Neuchatel, Switzerland (e-mail: aleksandar.jovanovic@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2022800

The rest of this paper is organized as follows: we survey the state of the art in FEC for streaming applications in Section II. Section III introduces the network, video, and FEC models. We discuss possible FEC and scheduling schemes for our proposed setup in Section IV. Algorithms for optimized multipath streaming are presented in Section V and evaluated in Section VI. Finally, we conclude the paper in Section VII.

II. Related Work

While media encoding with redundancy and error robustness/concealment features at the encoder/decoder offers some protection for the application against transmission failures [6]–[8], further network-layer protection mechanisms can be employed for improved robustness against network errors.

FEC and scheduling strategies lower the error probability for the transmitted packets at the expense of additional network resources and extra computations. Depending on the model for network losses [9], the application can adapt the FEC strategy [10]. Such action can be modeled as a joint source and channel coding (JSCC) optimization problem, whose purpose is to optimally allocate the network resources among media and redundant packets, so that the reconstructed quality of the media at the client is maximized. The authors of [11] deal with the optimal allocation of MPEG-2 encoding and media-independent forward error correction rates under the total given bandwidth. They define optimality in terms of minimum perceptual distortion given a set of video and network parameters. They compute the network error parameters after FEC decoding, and derive the global set of equations that lead to the optimal dynamic rate allocation. A similar analysis is performed in [12], where an optimal partitioning between byte-level FEC and packet-level FEC in the case of video multicast over wired and wireless networks is presented [13]. Finally, the authors of [14] present a performance analysis of packet-level FEC for streaming applications over congested network links.

Making a distinction among the media packets that need to be protected, the most advanced FEC strategies add more redundancy for the most important packets of the stream and less for the rest. UEP has been proved to better utilize network resources, enhancing thus the perceived quality of the multimedia application. Network adaptive error control schemes for video streaming using hierarchical FEC are presented in [15], [16]. Furthermore, the rate allocation and adaptation problem has been studied in simple one-path streaming scenarios. The authors of [17] propose a novel rate allocation scheme to be used with FEC in order to minimize the probability of packet loss in bursty loss environments such as those caused by network congestion. They present their protocols and compute the optimal rate allocation for the proposed distributed streaming model with FEC. Their work is later continued in [18] and [19].

All these works consider the network as a single transport link between the server and the client. They do not address multipath streaming scenarios, where more than one network path is allocated to the application. A more general JSCC (rate allocation) problem that takes into account intermediate active nodes or multiple existing paths between the server and one or more clients is presented in [20]. In this framework, intermediate overlay nodes can be used by a streaming application to perform specific tasks on the passing flow in order to improve the streaming process, e.g., traffic shaping and rate adaptation by video or FEC packet dropping. Also, the authors of [21] present a multicast streaming architecture in which intermediate nodes perform FEC coding/decoding operations on the stream in order to better cope with packet losses on the network links. A scheme for overlay multihop FEC for video streaming over peer-to-peer networks can be found in [22], while the authors of [23] discuss FEC protection and stream aggregation in intermediate nodes that forward multiple media flows over multiple network paths. However, none of these works address as a multipath streaming scenario, where the application has the choice on how to use the network resources (joint source channel rate allocation), and how to stream the required information (path selection and scheduling).

In this paper we present a study of different forward error correction and scheduling techniques for multipath scalable media streaming applications. We compare the most common error protection and scheduling strategies (e.g., equal error protection versus UEP, and earliest deadline first scheduling versus priority scheduling) in a JSCC framework, and we explore the tradeoff between computation complexity and optimality of results. We also explore the possible application of FEC codes in real systems where the choice of FEC modes is limited to a predefined and small set of parameters.

III. Multipath Streaming System

A. Network Model

We consider a framework where the multimedia streaming application uses a multipath network. The available network between the server S and the client C is modeled as a graph $G(V, E)$, where $V = \{N_i\}$ is the set of nodes in the network, and E is the set of links or segments (Fig. 1). In our analysis, we consider the graph G as a flow-equivalent graph, with the property that the maximum bandwidth offered to the application (e.g., the maxflow of G) does not depend on the choice of the transmission paths.[1] Flow-equivalent graphs contain every possible network graph that exhibits a single joint bottleneck network segment, or multiple joint bottleneck segments belonging to independent network subgraphs. More general network graphs may also belong to the category of flow-equivalent graphs, depending on the network segment parameters. Flow-equivalent graphs represent most common streaming scenarios, where the bottleneck links are generally shared by all paths as they lie on the last hop segment, or between Internet Service Providers (ISPs).

Each link $L_u = (N_i, N_j) \in E$ connecting nodes N_i and N_j has three associated positive metrics:

1) the available bandwidth $\rho_u > 0$ expressed in some appropriate unit (e.g., kb/s);
2) the average loss probability $\theta_u \in [0, 1]$, assumed to be an iid process, independent of the streaming rate;
3) the link propagation delay $t_u \geq 0$, considered as static.

[1] For a formal definition of flow-equivalent graphs, please refer to [24].

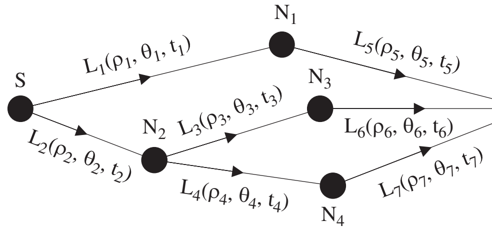
N$_1$
S
C
N$_3$
N$_2$
N$_4$
L$_1$(ρ$_1$, θ$_1$, t$_1$)
L$_5$(ρ$_5$, θ$_5$, t$_5$)
L$_3$(ρ$_3$, θ$_3$, t$_3$)
L$_6$(ρ$_6$, θ$_6$, t$_6$)
L$_2$(ρ$_2$, θ$_2$, t$_2$)
L$_4$(ρ$_4$, θ$_4$, t$_4$)
L$_7$(ρ$_7$, θ$_7$, t$_7$)

Fig. 1. Multipath streaming network scenario.

Finally, let $\mathcal{P} = \{P_1, \ldots, P_N\}$ denote the set of available loop-free paths between the server S and the client C in G, with N the total number of non-identical end-to-end paths [25]. A distinct path $P_i \in \mathcal{P}$ is characterized by the end-to-end bandwidth b_i computed as the minimum bandwidth of the intermediate network segments, the end-to-end loss probability p_i, computed as a multiplicative function of the individual loss probabilities of all segments composing the path and the end-to-end propagation delay τ_i, computed as the sum of the intermediate links delays

$$b_i = \min_{L_u \in P_i} (\rho_u), \tag{1}$$

$$p_i = 1 - \prod_{L_u \in P_i} (1 - \theta_u), \tag{2}$$

$$\tau_i = \sum_{L_u \in P_i} t_i. \tag{3}$$

The server S uses the available network paths for media packet transmission to the client. After initiating the media request, the client waits for a limited playback delay Δ before starting the playout.

B. Video Model

We represent the end-to-end distortion, as perceived by the media client, as the sum of the source distortion and the channel distortion. In other words, the quality depends on both the distortion due to a lossy encoding of the media information (D_S) and the distortion due to losses experienced in the network (D_L). Overall, the end-to-end distortion can thus be written as

$$D = D_S + D_L = f(R, \pi, \Gamma) \tag{4}$$

where Γ represents the set of parameters that describe the media sequence. This generic distortion model is quite commonly accepted, as it can accommodate a variety of streaming scenarios [11]. For example, when error correction is available, the total streaming rate has to be split between the video source rate that drives the source distortion D_S and the channel rate, which directly influences the video loss rate π.

We assume the video sequence to be layered encoded into L separate layers, each layer $l \leq L$ being characterized by its encoding rate r_l. Video layers are transmitted starting with the base layer, and then adding subsequent enhancement layers if the network conditions permit it. We assume that a video layer can either be fully transmitted or dropped from an encoder/sender point of view. Hence the total encoding rate of the video stream can be expressed as the sum of the rates of all layers that are transmitted from S to C

$$R = \sum_{j=1}^{l} r_j \tag{5}$$

where l is the number of transmitted video layers, as decided by the streaming application.

A commonly accepted model for the source rate–distortion is a decaying exponential function on the encoding rate, while the channel distortion is proportional in average to the number of lost pixels/video elements. Under the common assumption that network packets contains data referring to the same amount of video information (e.g., one frame, one slice, or one encoded video layer of a frame), the channel distortion is proportional to the number of lost packets, and is differentiated by the importance of the video layer containing the lost packets. For video encoding instances where higher video layers cannot be decoded unless all lower video layers are present at the decoder, we can explicitly formulate the video distortion metric as

$$D = \alpha \left(\sum_{j=1}^{l} r_j \right)^{\xi} + \beta \pi_1 + \sum_{j=2}^{l} \left(\pi_j (D_{j-1} - D_l) \prod_{s=1}^{j-1} (1 - \pi_s) \right) \tag{6}$$

where α, ξ, and $\beta \in \Gamma$ are sequence dependent parameters. D_j represents the source distortion of the first j layers of the video stream, and $\pi = \{\pi_j | \forall j : 1 \leq j \leq l\}$ is the set of average loss rates for the independent erasure processes experienced during the transmission of the video packets of each layer j. π_j depends on the loss probabilities p_i of the subset of network paths used for the transmission of the packets of video layer j and on the possible error protection scheme employed for protecting the video packets [26]. Notice that our model for the loss distortion D_L separates the packet losses in the base layer (seen as more severe, because of frame loss and the activation of error concealment strategies at the decoder) and the losses in the enhancement layers (seen as affecting only the total quality of the given frame). In our framework, we consider the packetized bitstream, with one network packet per frame and per video layer. Depending on available network resources, the server decides the number of video layers that can be transmitted to the client.

C. Forward Error Correction

Among all error correction techniques, packet-level FEC is generally preferred in the case of delay-sensitive or multicast-based streaming scenarios. Generically, a FEC block of n packets contains k media packets and $n - k$ FEC packets. Usually, the receiver can fully reconstruct the original k data packets as long as it correctly receives at least k packets of the FEC block.

We assume that the server S can protect each media layer against transmission errors, with one systematic forward error correction scheme $FEC(n, k)$. The loss probability for each video layer protected by $FEC(n, k)$ can be computed from

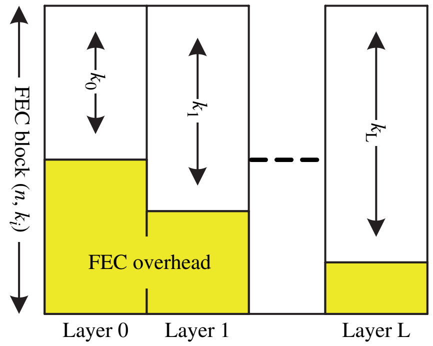

Fig. 2. UEP per video layer. Each video layer is protected by different FEC parameters irrespectively of the allocated transmission paths.

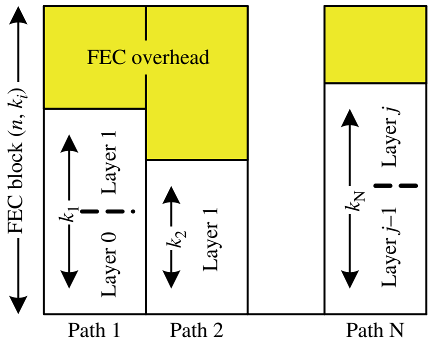

Fig. 3. UEP per network path. Each network path offers different FEC parameters for the protection of the data, no matter which video layer it belongs to.

the total error probability p affecting the transmission process of that layer. Let π_j be the error probability affecting video layer j after FEC decoding. It can be computed as the average probability of losing exactly i video packets from the FEC block $(1 \leq i \leq k)$ and at least $\lfloor n-k-i+1 \rfloor$ redundant packets

$$\pi_j = \frac{1}{k} \cdot \sum_{i=1}^{k} i \cdot e_i(n,k) \tag{7}$$

where $e_i(n,k)$ is the probability of losing at least $n-k+1$ packets from the FEC block, out of which exactly i packets are video packets. For an independent loss process, $e_i(n,k)$ can be easily computed [27] as

$$e_i(n,k) = \binom{k}{i} p^i (1-p)^{k-i} \sum_{l=\lfloor s+1-i \rfloor}^{s} \binom{s}{l} p^l (1-p)^{s-l} \tag{8}$$

where $s = n - k$.

Given the network and video models presented above, an upper bound on n can be computed as

$$n \leq f \cdot \min_{P_i \in \mathcal{P}} (\Delta - \tau_i) \tag{9}$$

where f is the encoded video sequence frame rate, Δ is the maximum playback delay allowed by the client, and P_i is an available end-to-end network path in $\mathcal{P}$. Knowing that the FEC performance in general increases with the increase in block size, we consider the maximum block size allowed by the network, e.g., $n = f \cdot \min_{P_i \in \mathcal{P}} (\Delta - \tau_i)$ as the FEC block size.[2]

IV. FEC and Scheduling Algorithms

We analyze in this paper several error control and packet scheduling strategies for effective multipath streaming. We study equal and unequal error protection and packet scheduling that possibly consider the importance of the video packets. Different algorithms are presented in this section. The three different error control algorithms and the two scheduling strategies are eventually combined to build six different streaming solutions. The performance of the different combinations of error control and scheduling algorithms are analyzed in the next sections.

A. Equal Error Protection Scheme

The first error control strategy studied in this paper is a simple equal error protection (EEP) scheme. All video layers are in this case protected by the same FEC scheme $FEC(n,k)$. Assume that each video layer $j \leq l$ is affected by the loss process p_j before FEC decoding at the client. The loss probability π_j after FEC reconstruction is computed based on the FEC parameters n, k, and p_j, according to (7) and (8). At the same time, the total rate of the video stream becomes

$$R = \sum_{j=1}^{l} r_j \cdot \frac{n}{k} \tag{10}$$

and is constrained by the total network available rate $\sum_{i=1}^{N} b_i$.

B. UEP Schemes

Next, we consider two cases of UEP when different video layers traversing different paths in the network can be protected by individual FEC schemes.

The first UEP scheme (UEP Layer) refers to video layers that are transmitted independently. In this case, each layer $j \leq l$ is protected by a separate FEC scheme $FEC(n, k_j)$ (Fig. 2). The total rate of video layer j becomes $r_j \cdot n/k_j$. The end-to-end loss process after FEC decoding π_j can be computed starting from p_j, n, and k_j, but it typically depends on the packet scheduling strategy.

The second UEP scheme (UEP Path) refers to individual network paths. All video data traversing a particular network path P_i is hence protected by a separate FEC scheme $FEC(n, k_i)$ (Fig. 3). We can compute the relevant end-to-end parameters of each path P_i in the network model (bandwidth $b_i^{'}$ and loss process $p_i^{'}$) by the values obtained after decoding the data protected by the FEC scheme $FEC(n, k_i)$. The available bandwidth for video packet transmission on path P_i becomes

$$b_i^{'} = b_i \cdot \frac{k_i}{n} \tag{11}$$

[2] While the complexity of the RS coding process grows as a quadratic function of n, in delay sensitive streaming scenarios, we expect n to be generally small, hence limiting the required coding execution time.

and the new loss process probability $p_i^{'}$ on path P_i can be computed from the FEC parameters n and k_i, and actual packet loss process p_i. Performing this transformation for every individual path $P_i \in \mathcal{P}$, we obtain an equivalent set of available network paths $\mathcal{P}^{'}$ for video streaming after FEC decoding. The new path parameters $b_i^{'}$ and $p_i^{'}$ affect the video flows according to the scheduling mechanism employed.

C. Earliest Deadline First Scheduling

Error control solutions have to be combined with packet scheduling solutions that are adapted to multipath streaming. We first consider a simple earliest deadline first (EDF) scheduling mechanism that is unaware of the characteristics of the network paths or of the specifics of the video encoding structure. The scheduling algorithm forwards the incoming media and FEC packets in a first-in-first-out (FIFO) order, on the first available network path, according to the respective rates and propagation delays. Note that this scheduling technique is tightly linked to the flow-equivalent network model described above. In the long run, the multimedia application will perceive the available network between S and C as one equivalent end-to-end network path with appropriate characteristics.

We can thus easily compute the parameters of the equivalent network end-to-end path, starting from the initial parameters of each individual network path P_i. Let b be the total bandwidth of the equivalent network model. In this case we can compute

$$b = \sum_{i=1}^{N} b_i. \tag{12}$$

The average loss probability p of the end-to-end equivalent network link can be computed as the average of the loss probabilities affecting each individual network path in $G(V, E)$

$$p = \frac{\sum_{i=1}^{N} b_i \cdot p_i}{\sum_{i=1}^{N} b_i}. \tag{13}$$

Finally, an upper bound on the propagation delay can be computed for the end-to-end equivalent network link as

$$\tau = \max_{i:1 \le i \le N} \tau_i. \tag{14}$$

Considering this scheduling mechanism, the transmitted video layers will experience the network as a single equivalent network path with the equivalent parameters as computed above. The maximum possible FEC block size n can be computed from the end-to-end propagation delay τ and Δ, while the error probability π_j affecting each video layer j, protected by a specific FEC code, can be computed from the loss probability p of the network link. Finally, the total source coding rate and FEC rate are upper-bounded by the total available bandwidth of the equivalent network link b.

D. Priority Scheduling

Finally, we consider a scheduling algorithm that takes into account the different parameters of the network paths and the relative importance of the video layers. As seen in [24], for flow-equivalent network graphs it is always best to fully utilize the network paths in ascending order of their loss probability p_i. Hence we adopt a scheduling strategy that maps the video layers, including the accompanying FEC rate, in increasing order of their importance, on the best available network paths in terms of loss probability.

Let $\mathcal{P} = \{P_1, \ldots, P_N\}$ be the ordered set of available network paths, according to their loss probabilities (e.g., $p_1 < \cdots < p_N$). Observe that two network paths P_i and P_j with equivalent error processes $p_i = p_j$ can be considered by the media application as a single network path with aggregated bandwidth $b_i + b_j$ and equivalent propagation delay $\max(\tau_i, \tau_j)$.

At the same time, let the l transmitted video layers be ordered according to their importance (e.g., layer 1 corresponds to the base layer, layer 2 corresponds to the first enhancement layer, etc.), and let $FEC(n, k_j)$ be the forward error correction scheme employed for protecting video layer $j \le l$. For simplicity reasons, we assume that the maximum FEC block size is computed in the same way as before. The total network rate required for the transmission of video layer j will be $r_j \cdot (n/k_j)$. We assume that layer j is mapped according to the priority scheduling algorithm described above on network paths $P_s, \ldots, P_t$ with corresponding rates $c_s, \ldots, c_t$, where $c_s \le b_s$, $c_t \le b_t$, and $c_i = b_i$, $\forall i : s < i < t$. We observe the following rate equality:

$$r_j \cdot \frac{n}{k_j} = \sum_{i=s}^{t} c_i \tag{15}$$

while the total error probability p_j affecting layer j before FEC decoding can be computed as

$$p_j = \frac{\sum_{i=s}^{t} c_i \cdot p_i}{\sum_{i=s}^{t} c_i}. \tag{16}$$

Based on p_j we can now compute the final error process affecting layer j after FEC decoding π_j according to (7) and (8). Observe that, compared to the previous scheduling case, where all transmitted video layers are affected by the same loss probability p, we schedule now the most important video layers on the best paths, hence we have $p_1 < \cdots < p_l$.

The FEC and scheduling mechanisms presented above can be combined pairwise to produce six different multipath streaming solutions, which are given in Table I. In the following sections, we compute the optimal rate allocation for each of these solutions and identify the best strategies for multipath media transmission and error correction.

TABLE I
DIFFERENT OPTIMIZATION ALGORITHMS FOR THE PROBLEM INSTANCES, BASED ON THE POSSIBLE COMBINATIONS OF SCHEDULING AND FEC STRATEGIES

	EEP	UEP Layer	UEP Path
FIFO Sch.	EqEEP	EqLayer	EqPath
Priority Sch.	SchEEP	SchLayer	SchPath

V. Optimization Algorithms

A. *Optimization Problem*

For each of the multipath streaming strategies described in Table I, we consider the problem of optimal rate allocation for a given video stream that can be split into flows sent on different paths from the streaming server S to the client C. Given the network rate constraints and path status in terms of propagation delay and loss probability, we are interested in finding the optimal rate split between source encoding rate and forward error protection rate in order to maximize the received video quality. Hence, we can formulate the optimization problem as follows:

Joint Multimedia—FEC Rate Allocation Problem (JMFR): Given the network graph G, the number of different paths or flows n, the video sequence characteristics (Γ), and the total number of encoded video layers L, find the optimal number of transmitted video layers l^*, and the optimal forward error protection scheme $FEC(n, k_j^*)$ for each layer $j \le l^*$, such that the perceived video distortion D at the client is minimized

$$\{l^*, k_j^*\} = \arg\min_{l \le L; k_j \le n; 1 \le j \le l} D(R, \pi, \Gamma) \qquad (17)$$

under the network rate constraint

$$\sum_{j=1}^{l^*} r_j \cdot \frac{n}{k_j^*} \le \sum_{i=1}^{N} b_i. \qquad (18)$$

The optimal rate allocation is *a priori* different for each streaming strategy in Table I. The optimization problem presents multiple instances depending on the FEC and scheduling mechanisms. Due to space limitations, we do not present every instance in detail here. But in each case, the rate and loss probability can be computed from the equations in the previous section. These values can later be used to compute the end-to-end distortion according to (6).

The optimization problem in each case can be solved by full search or dynamic programming solutions. It can also be noted that a Lagragian formulation can be used to recast the above problem as an unconstrained optimization problem. It might present an analytical solution lying on the convex hull of the rate–distortion characteristic in the scenarios where a closed-form expression can be written for the rate and end-to-end distortion. It is, however, not guaranteed that the optimal solution lies on the convex hull. We present and discuss the full search and the heuristic-based solutions in the rest of the section.

B. *Optimal Full Search Algorithms*

The full search algorithms are mostly used as a benchmark for the performance of the multipath streaming solutions. The algorithm finds the optimal solution for the optimization problem, by parsing every feasible rate allocation between source video rate and error correction rate. It outputs the optimal number of video layers to be transmitted, along with the optimal FEC strategy for each transmitted layer, such that the media distortion as perceived by the client is minimized.

Algorithm 1: EqLayer full search algorithm.

```
Input:
Network Graph $G(V, E)$, network paths
$\mathcal{P} = \{P_i(b_i, p_i, \tau_i) | \forall i : 1 \le i \le N\}$, encoded video
bitstream parameters $\Gamma$, video layers rates $r_l$,
$\forall l : 1 \le l \le L$, frame rate $f$, playback delay $\Delta$.
Output:
Optimal joint rate allocation $\{l^*, k_j^*\}$.
Initialization:
Compute equivalent link bandwidth: $b = \sum_{i=1}^{N} b_i$;
Compute equivalent link loss process: $p = \frac{\sum_{i=1}^{N} b_i \cdot p_i}{\sum_{i=1}^{N} b_i}$;
Compute equivalent link propagation delay: $\tau = \max_i \tau_i$;
Compute maximum FEC block size: $n = f \cdot (\Delta - \tau)$;
Procedure Compute optimal JMFR solution:
for Every number of video layers $l \le L$ and every
$k_j \le n$, $1 \le j \le l$ do
    Check rate constraint:
    if $\sum_{j=1}^{j} r_j \cdot \frac{n}{k_j} \le b$ then
        Compute $\pi_j$, $\forall j : 1 \le j \le l$, starting from $p$ and
        $k_j$;
        Compute $D = D(R, \pi, \Gamma)$ according to the
        equivalent network scheduling and UEP per video
        layer schemes;
    end if
end for
Output $\{l^*, k_j^*\} = \arg\min_{l \le L; k_j \le n; 1 \le j \le l} D(R, \pi, \Gamma)$.
```

For the sake of clarity, we present in Algorithm 1 the pseudo-code for one of these algorithms. The design of all others follow closely the same guidelines. Slight variations in the code will lead to the implementation of full search algorithms for all other streaming strategies.

While the algorithm outputs the optimal result for every network scenario, the computational complexity is rather high. During the full search for the optimal parameters, the algorithm needs to compute one distortion value for every feasible value of $k_j \le n$, for every video layer $j \le L$. Hence, the total complexity of the algorithm is $O(n^L)$. Similarly, the FEC strategy that allocates one FEC code per each individual network path requires a total of $O(n^N)$ computations, with N being the number of distinct available network paths. The exponential complexity of these algorithms will prohibit their use in large-scale scenarios with a large number of available network paths and fine granularity in the video encoding. Therefore, we introduce now heuristic algorithms that achieve suboptimal yet close results with a much lower computational complexity.

C. *Utility-Based Heuristic Algorithms*

In this section we introduce a suboptimal heuristic approach for solving the rate allocation problem. We build on the utility framework presented in [28], and develop algorithms that iteratively take a stepwise locally optimal decision.

Let each algorithm start from an initial feasible solution where only the video base layer, without any FEC protection, is scheduled for transmission, according to the employed scheduling mechanism. Let also $\mathcal{F}^s = \{l, \{k_j\}; 1 \le j \le l\}$ be a feasible solution obtained by our algorithms at iteration s.

We associate to this solution the total video rate $R^s = \sum_{j=1}^{l} r_j$ satisfying the total network rate constraint $\sum_{j=1}^{l} r_j \cdot n/k_j \le \sum_{i=1}^{N} b_i$. We can also compute the values $\pi^s = \{\pi_j; 1 \le j \le l\}$ representing the loss process observed by every transmitted video layer on the network. Based on these values, we can compute the perceived client distortion $D^s = D(R^s, \pi^s, \Gamma)$. Let B_s be the residual available network rate after transmitting all data packets related to solution $\mathcal{F}_s$.

At the next algorithm iteration $s+1$, we can either attempt the transmission of an extra video layer $l+1$, in case $l+1 \le L$, or change the FEC parameter k'_j of any of the already scheduled video layers $j \le l$. Let the new distortion measures associated to each of these actions be D^a_{s+1}, where a identifies the specific action taken. We define the utility of an action a as the ratio between the perceived video quality improvement by performing this action and the amount of network resources δr^a necessary for implementing the action

$$U_a = \frac{D_s - D^a_{s+1}}{\delta r^a}. \tag{19}$$

δr^a can be easily computed as r_{l+1} in case a new video layer is scheduled for transmission, or as the extra necessary network rate in order to change the FEC parameters of video layer j from k_j to k'_j, e.g., $\delta r^a = r_j n (k_j - k'_j)/(k_j k'_j)$. Any of the actions a is feasible as long as $\delta r^a \le B_s$. In the same time, action a brings an improvement in quality if $U_a > 0$.

The algorithm, at each iteration s, will choose the next solution $\mathcal{F}_{s+1}$ by performing the action that maximizes the utility value among all feasible actions. The algorithm stops either when there are no more feasible actions, e.g., the network rate has already been totally utilized, or there are no more actions that bring a positive improvement to the current solution. Depending on the FEC and scheduling mechanisms employed, six different algorithms can be derived. Algorithm 2 presents the pseudo code of one of them, the modifications toward all the others being straightforward.

For a complete search over the FEC parameter space, during each action a the parameter k'_j becomes $k'_j = k_j - 1$. In real system implementations, where only a limited amount of FEC schemes are available, k'_j should be chosen as the next smaller parameter from the feasible set of schemes after k_j.

During each iteration, the algorithm needs at most L computations, as we need to compute the utility of adding one extra video layer to the transmission or of changing by 1 the FEC parameter k_j of any of the already scheduled video layers j. At the same time, the maximum number of iterations is $n \cdot L$, as the algorithm terminates in the worst case when all video layers are scheduled with FEC parameters $k_j = 1$, $\forall j \le L$. Hence the total complexity of the proposed algorithm is $O(n \cdot L^2)$. In the following sections we assess the performance of our heuristic method compared to the optimal full search.

Algorithm 2: SchPath utility algorithm.

```
Input:
Network Graph $G(V, E)$, network paths
$\mathcal{P} = \{P_i(b_i, p_i, \tau_i) | \forall i : 1 \le i \le N\}$, encoded video
bitstream parameters $\Gamma$, video layers rates $r_l$,
$\forall l : 1 \le l \le L$, frame rate $f$, playback delay $\Delta$.
Output:
Optimal joint rate allocation $\{l^*, k^*_j\}$.
Initialization:
Compute maximum FEC block size:
$n = f \cdot \min_i(\Delta - \tau_i)$;
$\mathcal{F}_1 = \{1, k_1 = n\}$;
Compute $B_1 = \sum_{i=1}^{N} b_i - r_1$;
Compute the ordered set $\mathcal{P} = \{P_i : 1 \le i \le N\}$, s.t.
$p_1 <, \ldots, p_N$.
Procedure Compute heuristic JMFR solution:
Iteration s=1;
while 1 do
    for every feasible action $a$ do
        Compute updated distortion value $D^a_{s+1}$ according
        to the Priority Scheduling mechanism and UEP
        scheme;
        Compute utility function $U_a$;
    end for
    if no feasible action $a$ exists, or $U_a \le 0, \forall a$ then
        Break;
    end if
    Compute new solution: $\mathcal{F}_{s+1} = \arg\max_a U_a$;
    Update available network bandwidth $B_{s+1}$;
    Update iteration: $s = s + 1$.
end while
Output $\mathcal{F}_s$.
```

VI. Experimental Results

A. Setup

We test the proposed mechanisms in various network setups with various encoded bitstreams. We use a concatenated versions of the *foreman_cif* and *mobile_cif* sequences (3000 frames), encoded at 30 frames per second using the scalable encoder H.264/SVC. We encode the sequences in several video layers, one base layer, and one or more enhancement layers, at different encoding rates given by the chosen quantization parameters (QP). Our specific encoder generates the desired number of enhancement layers starting from the given QP value for the base layer and decreasing it by 6 for each additional layer. We assume that the video layers cannot be decoded unless all lower layers are available at the decoder.

We use a multipath network scenario that offers a variable number of end-to-end transmission paths to the media application. Our results are obtained for network scenarios with two, three, or four network paths. Unless otherwise stated, each network path is characterized by a random iid loss process with a loss probability that is randomly drawn in the interval 1–25%. The propagation delay is randomly selected in the interval 50–100 ms. The end-to-end bandwidth of each path is

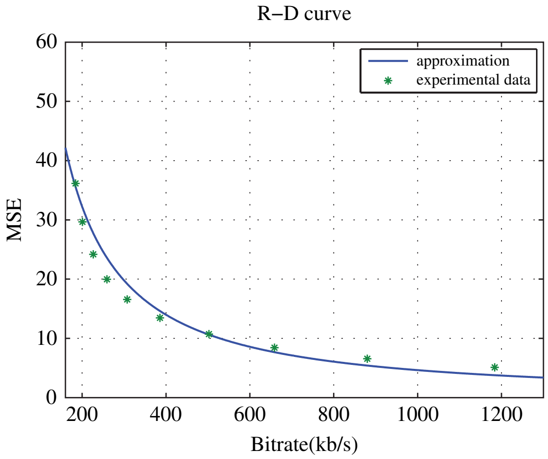

Fig. 4. Video model validation. Source distortion: H264/SVC encoder, *foreman_cif*, 30 frames/s, one BL and one EL, $\alpha = 1.9114 \times 10^4$, $\xi = -1.20515$.

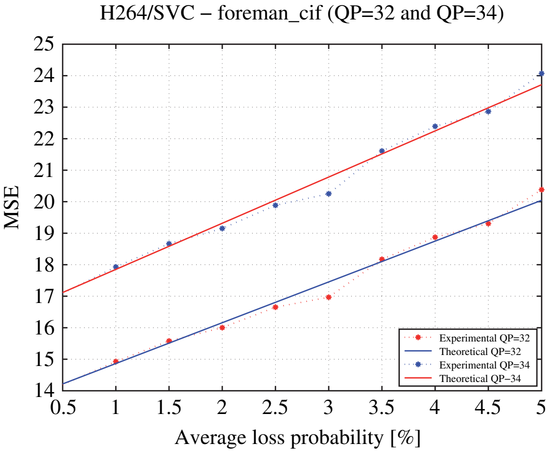

Fig. 5. Video model validation. Loss distortion: H264/SVC encoder, *foreman_cif*, 30 frames/s, one BL and one EL, $\beta = 147$.

randomly assigned in intervals that are meaningful for each experiment. Finally we assume that the client imposes a fixed playback delay $\Delta = 700\,\text{ms}$, after which it starts playing the received video data. Any packet arriving at the client after the decoding deadline is considered as lost for the application and discarded.

We start by validating the proposed distortion model for layered video streaming. Then, within the presented framework we compare the performance obtained by the proposed algorithms for optimal joint source-FEC rate allocation, representing the different FEC schemes and scheduling mechanisms presented above. Our results are averaged over 100 simulation runs for each network scenario and each transmitted bitstream. In particular, we emphasize the better performance brought by the UEP error correction scheme and the priority scheduling mechanism. Finally, we discuss real system implementations with constraints on the available set of FEC parameters.

TABLE II

OPTIMAL DISTORTION MODEL PARAMETERS

Parameter	*foreman_cif*	*mobile_cif*
α	$1.9114 \cdot 10^4$	$1.6668 \cdot 10^4$
ξ	-1.20515	-1.1510
β	147	346

B. Distortion Model Validation

We validate the distortion model with streaming experiments. We encode the *foreman_cif* and *mobile_cif* sequences (3000 frames, 30 frames per second) in one base layer (BL) and one enhancement layer (EL). The total rate of the encoded sequence is varied by encoding at different QPs for the BL. On the sequence of packets we are inflicting transmission packet losses according to an independent loss probability $p \in [0, 0.05]$, and we compare the decoded video quality with the original one by averaging over 100 simulation runs. Results for the validation of the source distortion are presented in Fig. 4, while Fig. 5 presents the validation of the loss distortion model (*foreman_cif* sequence). We observe that the model closely follows the experimental results. Similar results have been observed for the *mobile_cif* sequence. The optimal distortion model parameters obtained by fitting the experimental points to the theoretical distortion function are presented in Table II.[3]

C. EEP versus UEP

First we compare the EEP and UEP FEC schemes in the case of full search algorithms. We identify five network scenarios, ranging from very low end-to-end loss probability to very high one, and we set the end-to-end available bandwidth to be lower than the total encoded rate of the transmitted video bitstream. Each algorithm runs on the network scenario and optimizes the encoding FEC rate allocation in order to minimize the video distortion. They determine how many video layers to transmit and how much error protection to add to each layer, given the total network resource constraints.

Results for the two encoded sequences are presented in Figs. 6 and 7 for network rates that allow the scheduling of around two video layers. We observe that for every range of network losses, the UEP scheme performs better than the EEP scheme.[4] While the improvement is minimal for very low error network scenarios, it becomes increasingly visible as the network conditions get worse. For very high network error rates, all schemes fail to obtain good results, especially for the *mobile_cif* sequence that is much more sensitive to packet erasures. Similar results have been observed for different encoding rates. These results clearly prove the importance of flexible error protection in the case of scalable video transmission over lossy networks. The UEP scheme

[3] For a complete validation of the video distortion model, see [29].

[4] Observe that results for the EqPath and SchEEP schemes are similar in all our results. This is because from an implementation point of view, the two schemes are identical.

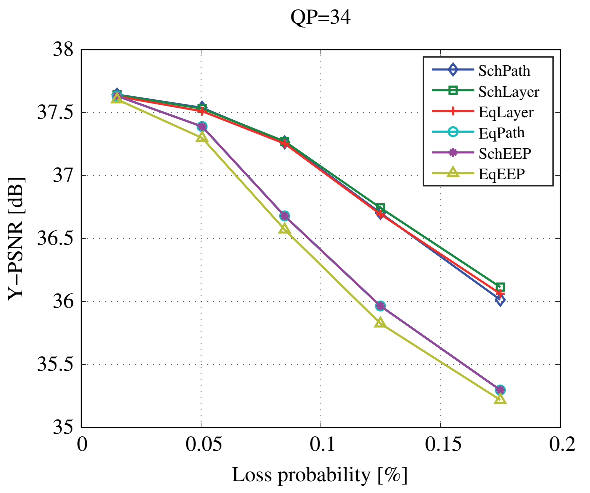

Fig. 6. FEC schemes comparison for various scheduling mechanisms, video base layer encoding: QP = 34 (*foreman_cif* sequence).

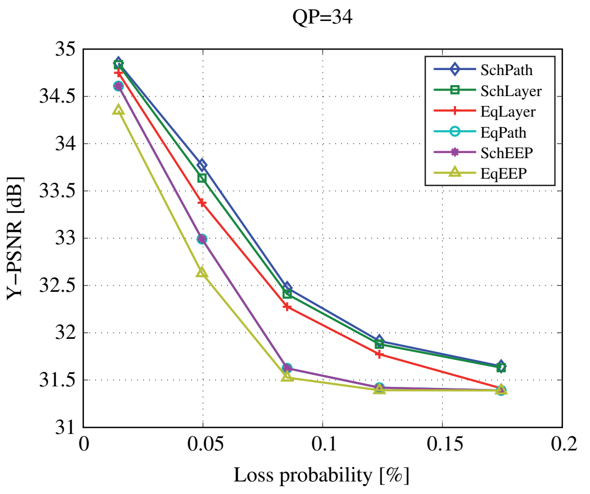

Fig. 7. FEC schemes comparison for various scheduling mechanisms, video base layer encoding: QP = 34 (*mobile_cif* sequence).

protects differently the video layers according to their overall importance to the final distortion measure and is able to better utilize network resources. On the other hand, the EEP scheme overprotects the higher layers of the video stream, hence wasting the available bandwidth.

Table III provides a different representation of the same results. Here we show the total error process associated with each transmitted video layer after FEC decoding at the client in the case of the UEP scheme. We observe that while the base layer is very well protected, ensuring practically zero losses, the higher layers are gradually less protected, as the application can tolerate a higher amount of losses with lower impact on the reconstructed media quality. We also observe that the equivalent network model performs the worst, as it fails to schedule higher layers of the video bitstream. On the other hand, the EEP scheme does not offer this flexibility, hence leading to a suboptimal performance.

TABLE III

AVERAGE LOSS RATE (IN %) AFTER FEC DECODING FOR EACH VIDEO LAYER, FOR THE ALGORITHMS BASED ON UEP (*foreman_cif* SEQUENCE)

	SchLayer	EqLayer	SchPath
Base layer	0.059	0.056	0.06
Enhancement layer 1	4	3.46	2.1
Enhancement layer 2	11.52	—	9.3

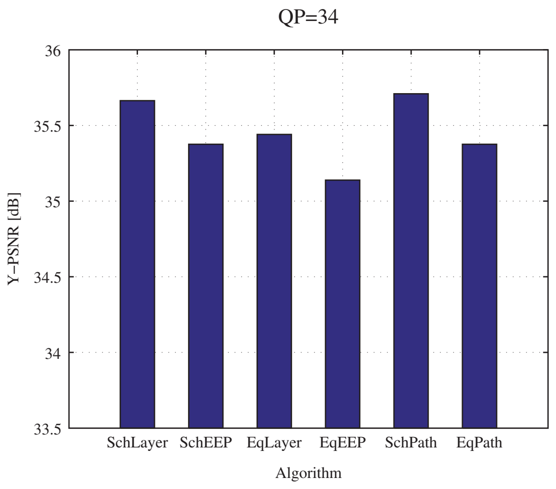

Fig. 8. Scheduling mechanisms comparison for various FEC strategies, video base layer encoding: QP = 34 (*foreman_cif* sequence).

D. EDF versus Priority Scheduling

Next, we compare the two proposed scheduling mechanisms. Due to the coarse granularity provided by the video encoder, in this section we hand-pick the network total bandwidth, such that we emphasize the conceptual differences between the two scheduling mechanisms.[5] We choose network scenarios with total end-to-end bandwidth that can easily accommodate the first two video layers of each bitstream without error protection (but not three layers), while randomly choosing the error rates of each path as presented before.

Figs. 8 and 9 present the obtained PSNR results for the proposed algorithms. Similar results have been obtained for both sequences encoded at different QPs. We observe that in general the priority scheduling with UEP performs better than the equivalent network scheduling, for all tested bitstreams. It can also be noted that all algorithms based on UEP outperform the EEP schemes. This corresponds to the results presented in the previous section. The difference in performance between the two scheduling mechanisms can be explained by the better resources utilization of the priority scheme. As the priority scheduling scheme sends the most important video layers on the better network paths in terms of error probability, it requires less rate for the error protection, hence being able to send more video layers. On the other hand, the equivalent network scheduling scheme considers

[5] Note that with fully scalable encoding systems, e.g., FGS encoders, the difference between the scheduling mechanisms would always be visible.

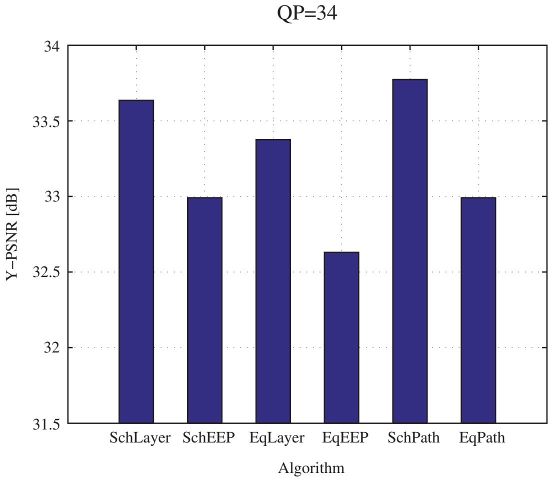

Fig. 9. Scheduling mechanisms comparison for various FEC strategies, video base layer encoding: QP = 34 (*mobile_cif* sequence).

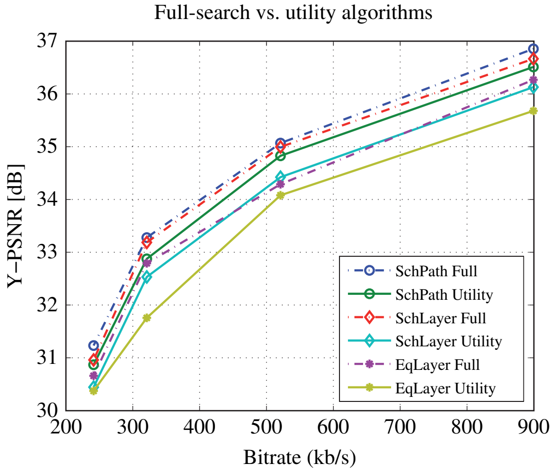

Fig. 10. Full search and utility-based algorithms performance for different video encoding rates (*foreman_cif* sequence).

TABLE IV
AVERAGE NUMBER OF TRANSMITTED VIDEO LAYERS FOR UEP-BASED ALGORITHMS IN VARIOUS NETWORK SCENARIOS (*foreman_cif* SEQUENCE)

	SchLayer	EqLayer	SchPath
Four-path scenarios	1.6	1.15	1.62
Three-path scenarios	1.55	1.23	1.53
Two-path scenarios	1.6	1.18	1.55

TABLE V
ALGORITHM PERFORMANCE IN SYSTEMS SCENARIOS WITH LIMITED CHOICE OF FEC PARAMETERS, AND PERCENTAGE OF TOTAL NETWORK RESOURCES UTILIZED (*foreman_cif* SEQUENCE)

	SchLayer	EqLayer	SchPath
Full search distortion (PSNR)	35.55	35.18	35.64
Utility distortion (PSNR)	35.02	35.03	34.04
Utility resource utilization (%)	80.17%	79.72%	91.36%

the network as a single equivalent link with equivalent error parameters, and hence requires more rate for the error protection of the most important layers. In turn, this leaves less resources for transmitting extra video layers. Table IV presents the average number of video layers transmitted by each of the algorithms utilizing UEP. We observe that, in general, the priority scheduling mechanisms manage to transmit more video information than the equivalent network mechanism on similar network setups.

E. Full Search versus Utility Algorithms

We compare the performance of the utility-based rate allocation algorithms to the full search ones. On the same network setups, we run both the full search and utility algorithms for bitstreams encoded at various bitrates. Fig. 10 presents the averaged PSNR results for the priority scheduling mechanisms. We observe that the heuristic utility-based algorithms have a performance that is similar to the one of the full search. At the same time, even for the simple scenarios considered here, with a limited number of network paths and video layers, and a small FEC block size, the utility-based algorithms require a much smaller computation effort.

The good performance of the heuristic algorithms is naturally motivated by the assumptions we have made on the encoding format (e.g., video layers are decoded in a sequential manner, and higher layers cannot be decoded unless previous layers have already been decoded). In addition, the previous results that illustrate the benefits of the unequal error protection based on the importance of each video layer further supports the idea of utility-based solutions.

Next, we consider the performance of real systems where the choice of FEC codes is limited to a finite set of parameters. Let the sender be able to access any of the following FEC codes: RS(20,16), RS(20,12), and RS(20,8) in order to protect the transmitted media packets. We test the utility-based algorithms constrained by the available set of FEC codes, and we compare the obtained results to the optimal ones found by the full search. Table V summarizes the results averaged over 100 simulation runs for one video bitstream transmitted over a hand-picked network scenario that emphasizes the differences between algorithms.

Compared to previous results, we observe a slight degradation in algorithm performance compared to the optimal full search results. This is explained by the reduced flexibility in the choice of the FEC mode. At the same time, we observe that full utilization of network resources is no longer optimal, due to the low granularity of the scheme. Depending on the algorithm, only a fraction of the network bandwidth is utilized in order to achieve the optimal result. We also observe that in the case of the *SchPath* algorithm, more video layers are scheduled for transmission in average, hence a higher bandwidth utilization. This comes, however, at the expense of a lower FEC protection of the lower layers (and degraded video quality), due to insufficient network resources

TABLE VI
COMPUTATION COMPLEXITY (NO. OF ITERATIONS) FOR THE THREE DISCUSSED APPROACHES: FULL SEARCH, UTILITY-BASED, AND LIMITED PARAMETER SET (*foreman_cif* SEQUENCE)

	EqLayer	SchLayer	SchPath
Full search	104 976	104 976	104 976
Utility-based	9	18	94
Limited-set	7	12	19

to accommodate a better FEC scheme from the available ones.

We conclude that flooding the network with data and redundant packets is not optimal, unless the designed system has full flexibility in the choice of FEC and scheduling strategies. This important result is in line with other findings concerning efficient media rate allocation in flow networks, as presented in [25].

F. Complexity Comparisons

Finally, we asses the complexity of the proposed utility-based mechanisms, compared to the benchmark full search algorithms. We compare the average total number of distortion-computation iterations required by each algorithm in order to output the solution for the optimization problem. Since during one iteration the algorithms compute one distortion value for a given feasible parameter set, and this computation is identical for all algorithms, analyzing the total number of iterations for each algorithm represents a fair comparison. Table VI summarizes our results for the three best performing algorithms (EqLayer, SchLayer, and SchPath) for the full search, utility, and limited parameter set cases. Our evaluation is conducted over the same scenario set as in Section VI-E.

We observe the huge complexity reduction offered by the proposed heuristic algorithms compared to the full search optimal benchmark solution. While all full search algorithms have the same complexity, as they have to search through the complete parameter space in the given scenario, the heuristic algorithms complexity depends on the specific allocation strategy. In general, the simplest scheme requires the smallest number of iterations, but provides worse results, while the complexity of the UEP schemes with priority scheduling depends on the total number of available paths and scheduled layers. In large network scenarios with more available paths for transmission, the priority scheduling scheme with UEP for the video layers is faster, while the UEP scheme based on the network paths offers more granularity and possibly improved results.

VII. CONCLUSION

We address the problem of optimal joint source-channel rate allocation for multimedia streaming applications over lossy multipath networks. Based on different FEC and scheduling strategies for layered encoded video streaming, we derive algorithms for the efficient computation of the source rate and FEC rate, with the final goal of optimizing the client-perceived video quality. In a lossy multipath scenario with limited network resources, we find optimal to perform a prioritized scheduling of the video layers according to their importance on the best network paths first. At the same time, UEP strategies that protect better the most important video information are shown to be the most efficient error control solutions. With a minimal computation cost at the server side, these strategies improve the received quality of the transmitted video compared to simpler strategies that do not take into account the network topology or the video encoding format. Our heuristic source-channel rate allocation method based on utility functions is shown to provide close to optimal results, avoiding the computational cost of full search optimizations. We also discuss real system implementations when the optimization problem is solved only on an available set of video rates and FEC strategies. We show that in such a case, flooding all available network paths is no longer optimal in terms of reconstructed media quality, and that a tradeoff between the total streamed rate and available network conditions should be carefully considered.

REFERENCES

[1] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path versus multi-path overlay routing," in *Proc. 3rd ACM SIGCOMM Conf. Internet Measurements*, Oct. 2003, pp. 91–100.

[2] L. Golubchik, J. Lui, T. Tung, A. Chow, and W. Lee, "Multi-path continuous media streaming: What are the benefits?" *ACM J. Performance Eval.*, vol. 49, no. 1–4, pp. 429–449, Sep. 2002.

[3] F. Wu, H. Sun, G. Shen, S. Li, Y.-Q. Zhang, B. Lin, and M. C. Li, "SMART: An efficient, scalable and robust streaming video system," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 2, pp. 192–206, 2004.

[4] D. G. Sachs, I. Kozinetsev, M. Yeung, and D. L. Jones, "Hybrid ARQ for robust video streaming over wireless LANs," in *Proc. Int. Conf. Inform. Technol.: Coding Comput. (ITCC)*, Las Vegas, NV, 2001, pp. 317–321.

[5] G. B. Horn, P. Knudsgaard, S. B. Lassen, M. Luby, and J. E. Rasmussen, "A scalable and reliable paradigm for media on demand," *IEEE Comput. Mag.*, vol. 34, no. 9, pp. 40–45, Sep. 2001.

[6] B. Girod and N. Farber, "Feedback-based error control for mobile video transmissions," *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.

[7] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques, real-time video communications over unreliable networks," *IEEE Signal Process. Mag.*, vol. 17, no. 4, pp. 61–82, Jul. 2000.

[8] Y.-K. Wang, M. M. Hannuksela, and V. Varsa, "The error concelament feature in the H.26L test model," in *Proc. IEEE ICIP*, vol. 2, 2002, pp. II-729–II-732

[9] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: Does burst-length matter?" in *Proc. IEEE ICASSP*, vol. 5. Apr. 2003, pp. 684–687.

[10] P. Frossard, "FEC performance in multimedia streaming," *IEEE Commun. Lett.*, vol. 5, no. 3, pp. 122–124, Mar. 2001.

[11] P. Frossard and O. Verscheure, "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," *IEEE Trans. Image Process.*, vol. 10, no. 12, pp. 1815–1825, Dec. 2001.

[12] Y.-C. Su, C.-S. Yang, and C.-W. Lee, "Optimal FEC assignment for scalable video transmission over burst error channel with loss rate uplink," *Signal Process.: Image Commun.*, vol. 18, no. 7, pp. 537–547, Aug. 2003.

[13] T.-W. A. Lee, S.-H. G. Chan, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Optimal allocation of packet-level and byte-level FEC in video multicast over wired and wireless networks," in *Proc. IEEE Globecom*, vol. 3. San Antonio, TX, Nov. 2001, pp. 1994–1998.

[14] X. Yu, J. Modestino, and I. Bajic, "Performance analysis of the efficacy of packet-level FEC in improving video transport over networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP '05)*, pp. II-177–II-180.

[15] W.-T. Tan and A. Zakhor, "Error control for video multicast using hierarchical FEC," in *Proc. 6th IEEE Int. Conf. Image Process. (ICIP)*, vol. 1. Kobe, Japan, 1999, pp. 401–405.

[16] Q. Zhang, G. Wang, W. Zhu, and Y.-Q. Zhang, "Robust scalable video streaming over internet with network-adaptive congestion control and unequal loss protection," in *Proc. Packet Video Workshop EURASIP/IEEE*, 2001.

[17] T. Nguyen and A. Zakhor, "Distributed video streaming over the internet," in *Proc. Multimedia Comput. Network. (MMCN)*, Jan. 2002, pp. 186–195.

[18] T. Nguyen, P. Mehra, and A. Zakhor, "Path diversity and bandwidth allocation for multimedia streaming," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, vol. 1. Jul. 2003, pp. 1–4.

[19] T. Nguyen and A. Zakhor, "Path diversity with forward error correction (PDF) system for packet switched networks," in *Proc. IEEE INFOCOM*, vol. 3. 2003, pp. 663–672.

[20] Y. Shan, I. Bajic, S. Kalyanaraman, and J. Woods, "Joint source network error control coding for scalable overlay video streaming," in *Proc. IEEE ICIP*, 2005, pp. I-177-80.

[21] H. Radha and M. Wu, "Overlay and peer-to-peer multicast with network-embedded FEC," in *Proc. IEEE ICIP*, vol. 3. Oct. 2004, pp. 1747–1750.

[22] Y. Shan, I. V. Bajic, S. Kalyanaraman, J. W. Woods, "Overlay multi-hop FEC scheme for video streaming over peer-to-peer networks," in *Proc. IEEE ICIP*, vol. 5. Oct. 2004, pp. 3133–3136.

[23] G.-M. Su, M. Chen, and M. Wu, "Cross-path pdma-based error protection for streaming multiuser video over multiple paths," in *Proc. IEEE ICIP*, Atlanta, GA, Oct. 2006, pp. 21–24.

[24] D. Jurca, "Adaptive media streaming over multipath networks," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, EPFL-STI-LTS4, Lausanne, Switzerland, Oct. 2007.

[25] D. Jurca and P. Frossard, "Media-specific rate allocation in multipath networks," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1227–1240, Oct. 2007.

[26] I. Radulovic and P. Frossard, "Adaptive video streaming in lossy networks: Versions or layers?" in *Proc. IEEE ICME*, vol. 3. Jun. 2004, pp. 1915–1918.

[27] D. Jurca and P. Frossard, "Optimal FEC rate for media streaming in active networks," in *Proc. IEEE ICME*, vol. 2. Taipei, Taiwan, Jun. 2004, pp. 1319–1322.

[28] F. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 2, pp. 5–12, Apr. 2005.

[29] A. Jovanovic, "Media aware rate allocation and FEC protection of streaming video in multipath networks," M.S. thesis, Ecole Polytechnique Fédérale de Lausanne, EPFL-STI-LTS4, Lausanne, Switzerland, Mar. 2007.

Dan Jurca (S'03–M'07) received the M.S. degree in telecommunications and computer networks from the "Politehnica" University of Timsoara, Romania, in 2002, and the Ph.D. degree in electrical engineering from the Ecole Polytechnique Féderale de Lausanne, Lausanne, Switzerland, in 2007, with his main topic of research focused on real-time adaptive multimedia streaming. From 2007 to 2008, he was a Postdoctoral Researcher in the Royal Institute of Technology, Stockholm, Sweden, researching distributed network management aspects.

He is currently a Researcher for DOCOMO Communications Laboratories GmbH, Munich, Germany, in the Ubiquitous Networking Research Group. His research interests focus on network virtualization, multimedia streaming, network coding, and network cross-layer design.

Pascal Frossard (S'96–M'01–SM'04) received the M.S. and Ph.D. degrees, both in electrical engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively. Between 2001 and 2003, he was a member of the Research Staff at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he was on media coding and streaming technologies.

Since 2003, he has been an Assistant Professor at EPFL, where he heads the Signal Processing Laboratory. His research interests include image representation and coding, nonlinear representations, visual information analysis, joint source and channel coding, multimedia communications, and multimedia content distribution. He was the recipient of the Swiss NSF Professorship Award in 2003, the IBM Faculty Award in 2005, and the IBM Exploratory Stream Analytics Innovation Award in 2008.

Dr. Frossard has been the General Chair of IEEE ICME 2002 and Packet Video 2007, and a Member of the Organizing or Technical Program Committees of numerous conferences. He has been an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA since 2004 and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY since 2006. He is an elected member of the IEEE Image and Multidimensional Signal Processing Technical Committee since 2007, the IEEE Visual Signal Processing and Communications Technical Committee since 2006, and the IEEE Multimedia Systems and Applications Technical Committee since 2005. He has served as Vice-Chair of the IEEE Multimedia Communications Technical Committee from 2004–2006 and as a member of the IEEE Multimedia Signal Processing Technical Committee from 2004–2007.

Aleksandar Jovanovic received the Diploma in telecommunication engineering from the Faculty of Electronic Engineering, Nis, Serbia, in 2004, and the M.Sc. degree in communication system from Ecole Politechnique Féderale de Lausanne (EPFL), Lausanne, Switzerland, in 2007. After graduation, he was in the Laboratory for Computer Communications and Applications at EPFL.

From September 2008, he has been pursuing the Ph.D. degree at the Institute of Management Technology at EPFL, The Electronics and Signal Processing Laboratory, Neuchatel in the area of Global Navigation Satellite Systems (GNSS), under the supervision of Prof. Piere-Andre Farine. His research interests include advanced algorithms for acquisition and tracking of GNSS, software receiver concept, multimedia networking, mobile communication systems and wireless networking.