Exploiting the Path Propagation Time Differences in Multipath Transmission with FEC

Maciej Kurant  
LCA - School of Communications and Computer Science  
EPFL, CH-1015 Lausanne, Switzerland  
Email: maciej.kurant@epfl.ch

Abstract—We consider a transmission of a delay-sensitive data stream from a single source to a single destination. The reliability of this transmission may suffer from bursty packet losses - the predominant type of failures in today’s Internet. An effective and well studied solution to this problem is to protect the data by a Forward Error Correction (FEC) code and send the FEC packets over multiple paths. In this paper we show that the performance of such a multipath FEC scheme can often be further improved. Our key observation is that the propagation times on the available paths often significantly differ, typically by 10-100ms. We propose to exploit these differences by appropriate packet scheduling that we call ‘Spread’. We evaluate our solution with a precise, analytical formulation and trace-driven simulations. Our studies show that Spread substantially outperforms the state-of-the-art solutions. It typically achieves two- to five-fold improvement (reduction) in the effective loss rate. Or conversely, keeping the same level of effective loss rate, Spread significantly decreases the observed delays and helps fighting the delay jitter.

I. INTRODUCTION

We consider a transmission of a delay-sensitive data stream from a single source to a single destination. How to improve the reliability of this transmission? Traditional ARQ (Automatic Repeat-reQuest) mechanisms often cannot be used, as they impose additional and usually unacceptable delays of at least one RTT (Round Trip Time). A more applicable technique is to introduce some type of redundancy, e.g., Forward Error Correction (FEC). Clearly, due to the delay constraints, a FEC block must be of limited length [1]. This, in turn, makes it inefficient against bursty packet losses [1] - the predominant type of losses in today’s Internet [2].

A good solution to this problem is to assign the FEC packets to *multiple paths* spanning the source and the destination [3]–[10]. An illustration of a multipath FEC system is presented in Fig. 1. Theoretically, the multiple paths could be constructed with the help of source routing, but this technique is not yet fully available in the Internet. A more practical alternative is the usage of overlay relay nodes that forward the traffic (as in Fig. 1). If the resulting paths are statistically independent, which is especially likely for multi-homed hosts, then the loss bursts get averaged out and FEC regains effectiveness. Similar performance benefits due to multipath were also observed in the context of Multiple Description Coding [11].

When designing a system that splits a FEC block across multiple paths, we have to (1) select some paths out of all candidates, (2) assign the transmission rates to these paths, and (3) schedule the packets. The previous studies proposed techniques to solve (1-3) as a function of the statistical loss properties of the paths [4,5,10].

However, there are other important parameters affecting the performance of the multipath FEC system. In particular, in this paper we show that the propagation times on the available multiple paths often significantly differ. These differences, in turn, can be exploited to improve the system reliability. Below, we explain and motivate our approach on concrete examples and measurements.

A. Propagation times on direct and indirect paths may differ significantly

In Fig. 2 we study the path propagation time differences in the real-life Internet. The measurements were collected by running all-to-all traceroutes between 326 nodes in DIMES [12]. These nodes are usually private hosts located at different sites around the world. (We obtained similar results for measurements on PlanetLab [13].)

For each source-destination pair we construct a set of $R$ paths. We always include the *direct* path $P_1$ with propagation time $t_1$. Each of the remaining $R-1$ paths is *indirect*, i.e., it uses some overlay relay node to forward the traffic. We choose uniformly at random a number $C$ of candidate relay nodes among the remaining 324 DIMES nodes. This results in $C$ candidate indirect paths. From them we select the $R-1$ indirect paths following the intuitive selection procedure given in [5]. For $R=2$ paths we choose the indirect candidate path that is the most IP link disjoint with the direct path $P_1$. Clearly, this minimizes the loss correlation between $P_1$ and $P_2$. If there are more paths that achieve the minimal IP overlap, then the
one with the smallest propagation time is kept. For $R > 2$ we proceed similarly, except that now we consider the aggregated values of IP overlap and propagation time, i.e., summed over all $R(R - 1)/2$ possible path pairs.

According to Fig. 2 for $R = 2$, the best indirect path $P_2$ has propagation times larger by typically $0 \ldots 75 \text{ms}$ than the direct path $P_1$ (see $t_2 - t_1$ in top-left histogram). This difference gets larger for a smaller number of candidates $C$ (table at the bottom).

Moreover, the path propagation time differences grow significantly with the number of paths $R$ used in the system. As shown in Fig. 2 already for $R = 3$ the medians of the distributions are roughly doubled compared to $R = 2$, and typically $P_3$ is faster than the slower of the two indirect paths by $\max(t_2, t_3) - t_1 \approx 0 \ldots 150 \text{ms}$.

We conclude that in the real-life Internet the propagation time differences on multiple paths between a source-destination pair are significant, typically reaching several tens of milliseconds.

### B. The differences in propagation times can be exploited by a multipath FEC system

We propose to exploit these path propagation time differences when designing a multipath FEC system. Our solution is easy to implement and can bring significant performance gains. Let us take a concrete example described in Fig. 3. Assume that there exist two paths between the source and the destination, the direct path $P_1$, and an indirect path $P_2$ created by employing another peer that works as a relay. Let $t_1 = 100 \text{ms}$ and $t_2 = 150 \text{ms}$ be the propagation delays on $P_1$ and $P_2$, respectively. So the path propagation time difference is $\Delta t = 50 \text{ms}$ (Fig. 3A). Let the two paths be lossy with the same loss rate $1\%$ and the same average loss burst length of $10 \text{ms}$, but independent. The data packets are generated at the source every $T = 5 \text{ms}$. If no form of packet protection is used, then the data packet loss rate observed at the destination, or the effective loss rate, is $\pi_B^* = 1\%$ (B). Assume now that we use FEC(6,4) to protect the packets. If we send all packets on $P_1$ with inter-packet times $T$, then the effective loss rate after FEC decoding is $\pi_B^* = 0.553\%$ (C). Following [5,10], we now split the packets equally between $P_1$ and $P_2$, which decreases $\pi_B^*$ to $0.148\%$ (D). This solution represents the state of the art. Note that now the last FEC packet on path $P_2$ reaches the destination $t_{\text{FEC}} = t_2 + 4 \cdot T = 170 \text{ milliseconds}$ after the generation of the first FEC packet at source. In other words, in this case the application using multipath FEC must accept the (maximal) delay equal to $t_{\text{FEC}}$. However, we can achieve far better results still respecting this delay constraint. For instance, we can appropriately increase the packet-spacing on $P_1$ and achieve $\pi_B^* = 0.113\%$ (E). Finally, we get even more significant improvement by sending 4 packets on $P_1$ and 2 packets $P_2$, i.e., by applying unequal rates on the paths (F). This results in $\pi_B^* = 0.016\%$, which is almost one order of magnitude smaller than (D).

In other words, we exploit the differences in path propagation times by spreading the packets in time, such that the maximal allowed delay is respected. The gain over the state of the art measured in the effective loss rate $\pi_B^*$ may be very significant (here $0.016\%$ vs $0.148\%$, i.e., almost tenfold). Moreover, some results may seem counterintuitive. For instance, it may be better to use only one path than to use two (un-spaced) paths. It also turns out that even if the loss distributions on the paths are the same, the optimal rates assigned to these paths are not necessarily equal.

---

Fig. 2. The difference between propagation times on the direct path $P_1$ and the best indirect paths $P_2$ and $P_3$. We present the results for $R = 2$ (two paths: one direct and one indirect) and $R = 3$ (the direct path and two indirect paths). The histograms (top) show the distribution of propagation time differences for $C = 5$ available candidate indirect paths. The table (bottom) shows the medians of these distributions for $C = 1, 5$ and $50$. The averages (not shown) are systematically higher than the medians.

| $R$ | $t_2 - t_1$ [ms] | $\max(t_2, t_3) - t_1$ |
|-----|-----------------|----------------------|
| 2   | 43              | N/A                  |
| 3   | 37              | 18                   |
| 50  | 69              | 39                   |

Fig. 3. Illustration of various packet schedules and their performance measured in the effective loss rate $\pi_B^*$. We use two independent paths $P_1$ and $P_2$ with identical failure distributions. The data packets are generated at the source every $T = 5 \text{ms}$ and coded with FEC(6,4). (A) The path propagation time $t_2$ on path $P_2$ is $\Delta t = t_2 - t_1 = 50 \text{ms}$ larger than the path propagation time $t_1$ on path $P_1$. (B) No FEC, single path, the packets are sent at times $0, 5, 10, 15 \text{ms}$. (C) FEC on $P_1$ only, packets are sent as soon as they are generated, i.e., we use the ‘Immediate’ schedule $S^\text{imm}$. (D) Packets alternate between $P_1$ and $P_2$ with equal rates $n_1 = n_2 = 3$, as in [5,10]. The total FEC block delay resulting from this scheme serves as a maximal FEC block delay in the following scenarios. (E) Packets alternate between $P_1$ and $P_2$ with equal rates, but the three packets sent on $P_2$ are maximally spread. (F) Packets are split between $P_1$ and $P_2$ with optimal rates $n_1 = 4, n_2 = 2$, maximally spread.
C. Organization of this paper

The remainder of this paper is organized as follows. In Section III we fully specify our model, which allows us to precisely state the problem we are solving. Next, in Section IV we derive exact analytical expressions for the effective loss rate $\pi_B^r$ under multipath FEC and an arbitrary schedule. In Section V we describe the ‘Immediate’ schedule representing the state of the art, and propose a ‘Spread’ schedule that exploits the differences in path propagation times. In Section VI we evaluate our solution analytically, by simulations and by trace-driven simulations fed with real-life Internet traces. In Section VII we discuss the related work. Finally we conclude the paper and propose future directions. The details of some calculations are put in Appendix.

II. MODEL AND PROBLEM STATEMENT

The packets, called data packets, are generated at source $s$, with constant inter-arrival time $T$. There exist $R$ paths between sender $s$ and destination $d$, with the propagation delays $t_1, \ldots, t_R$, respectively.

A. Path losses

The paths are assumed to be independent. We model bursty losses on each path by the popular two-state Gilbert model. Its basic version is a Discrete Time Markov Chain (DTMC), and captures the loss correlations due to queuing on bottleneck links, when the path is sampled at some constant rate (e.g., $1/T$). However, as we vary the sampling rates, DTMC is not sufficient. Indeed, on the same path we experience much higher loss burstiness under the packet interval of $T = 5 ms$ than of $T = 100 ms [1]$. For this reason we use the continuous-time version of the Gilbert model [4,14] that naturally accommodates different sampling rates. It is a two-state stationary Continuous Time Markov Chain (CTMC) $\{X_r(t)\}$. The state $X_r(t)$ at time $t$ assumes one of the two values: $G$ (‘good’) or $B$ (‘bad’). If a packet is sent at time $t$ and $X_r(t) = G$ then the packet is transmitted; if $X_r(t) = B$ then the packet is lost.

We denote by $\pi_G^r$ and $\pi_B^r$ the stationary probabilities that the $r$th path is good or bad, respectively. Similarly, let $\mu_G^r$ and $\mu_B^r$ be the transition rates from $G$ to $B$ and from $B$ to $G$, respectively. In this paper we use two meaningful, system-dependent parameters to specify the CTMC packet loss model:

- the average loss rate $\pi_B^r$; and
- the average loss burst length (in seconds) $1/\mu_B^r$.

All other parameters can be easily derived from these two, because

$$\pi_G^r \text{ and } \pi_B^r = \frac{\mu_B^r}{\mu_G^r + \mu_B^r}. \quad (1)$$

B. Multipath FEC

We use a $\text{FEC}(n,k)$ scheme to protect the data packets against losses (see Fig. 1). This means that $k$ data packets (not necessarily consecutive) are encoded as one FEC block of $n$ packets, called FEC packets. In particular, as in [4,6,10,15,16], we consider a systematic FEC, i.e., a scheme where the first $k$ packets are the $k$ data packets (unchanged). The remaining $n-k$ packets, called redundancy packets, carry the redundancy information. The destination uses the redundancy packets to recover some of the lost data packets as follows. Let $F$ be the number of lost FEC packets and let $D$ be the number of lost data packets of a FEC block, both before the FEC recovery (note that $D$ contributes to $F$). If $F \leq n-k$ then all the $n$ FEC packets and hence all the $k$ data packets are recovered. In contrast, if $F > n-k$, then no FEC recovery is possible and $D$ data packets are lost.

C. Packet scheduling

Finally, the packets are sent according to some schedule that defines when and on which path each FEC packet is sent. More precisely, we denote by $\mathcal{S} = (T,R)$ the schedule of packets in a FEC block, where $T$ and $R$ are vectors of length $n$. The

| $F$, $E$ | probability, expected value |
| $s$ | source node |
| $d$ | destination node |
| $T$ | (constant) interval between two consecutive data packets at source $s$ |
| $R$ | number of independent paths between source $s$ and destination $d$ |
| $P_r$ | $r$th path |
| $t_r$ | propagation delay on $P_r$ |
| $\pi_B^r$, $1/\mu_B^r$ | the average loss rate and loss burst length on path $P_r$ |
| $n, k, (n-k)$ | number of FEC packets assigned to $P_r$ (rate of path $P_r$) |
| $k_t$ | (constant) spacing of the $n_t$ packets on path $P_t$ |
| $T_F$, $D$ | number of lost FEC and data packets before FEC recovery |
| $\pi_B^s$ | effective loss rate, i.e., the expected fraction of lost data packets |
| $t_{\text{FEC}}$ | FEC block transmission time, i.e., the time between the generation of the first FEC packet at source $s$ and the scheduled delivery of the latest FEC packet at destination $d$ |

Fig. 4. An illustration of a schedule $\mathcal{S} = (T,R)$ on $R = 2$ paths with FEC(6,4). Four data packets numbered 1-4 are generated at the source at equal intervals $T$; the first one specifies time $t = 0$. The $n-k = 2$ redundancy packets are numbered 5 and 6. According to the schedule $\mathcal{S} = (T,R)$, the $i$th FEC packet is sent at time $T(i)$ over path $R(i)$.
ith FEC packet is sent at time $T(i)$ over path $R(i)$, as shown in Fig. 4. The time is counted from the generation (at the source) of the first data packet of the FEC block. Denote by $t_{\text{FEC}}$ the FEC block transmission time, i.e., the time between the generation of the first FEC packet at source $s$ and the scheduled delivery of the latest FEC packet at destination $d$. Given a schedule $S$, $t_{\text{FEC}}$ can be easily computed as

$$t_{\text{FEC}} = \max_{1 \leq i \leq n} \left( T(i) + t_{R(i)} \right).$$

(2)

For a given schedule, $t_{\text{FEC}}$ can be interpreted as the total delay imposed by the multipath FEC system on the delay-sensitive application using it. Indeed, if the first packet of a FEC block is lost and needs to be reconstructed by FEC, then we have to wait up to $t_{\text{FEC}}$ until the destination is reached by the other FEC packets necessary for the reconstruction of the lost packet. In practice, however, a constraint is likely to come from the delay-constrained application itself, as the maximal acceptable delay $t_{\text{FEC}}$. In this case our goal is to design a good schedule respecting this constraint, which is the approach used in this paper.

The schedule also implicitly defines the rate $n_r$ of path $P_r$, i.e., the number of FEC packets sent on $P_r$. Similarly, let $k_r$ be the number of data packets among the $n_r$ packets sent on $P_r$. Clearly, $\sum_r n_r = n$ and $\sum_r k_r = k$.

D. Effective loss rate $\pi^*_B$ and problem statement

Our ultimate goal is to send a stream of data packets over (possibly multiple) lossy channels in a way that minimizes the losses observed at the destination, given a maximal value for $t_{\text{FEC}}$. Therefore, we adopt a natural performance metric called effective loss rate $\pi^*_B$. It is defined as the expected fraction of lost data packets observed at the destination after an attempt of FEC decoding. Now the problem can be stated as follows:

Given the path loss properties $(\pi^{(r)}_G, 1/\mu^{(r)}_G)$ and $t_r$ for every path $P_r$, the FEC parameters $(n$ and $k)$ and maximal FEC block transmission time $t_{\text{FEC}}$, find the schedule $S$ that minimizes the effective loss rate $\pi^*_B$.

We approach this problem in two steps. First, in Section III we derive an exact analytical formula for the effective loss rate $\pi^*_B$ for a given schedule $S$. Second, in Section IV we introduce a schedule that exploits the differences in path propagation times and outperforms the schedules proposed to date.

III. EXACT ANALYTICAL DERIVATION OF THE EFFECTIVE LOSS RATE $\pi^*_B$

In order to design a good schedule we must be able to evaluate it. In this section we derive the exact analytical expression for the effective loss rate $\pi^*_B$ for a given schedule $S$. We consider two cases. First, we derive $\pi^*_B$ for an arbitrary schedule $S$. The resulting formula is simple but computationally expensive and untractable for larger sizes $n$ of the FEC block. Next, we derive $\pi^*_B$ assuming that on each path separately the packets are evenly spaced. This constraint is compatible with the schedule we propose later and results in a computationally lighter formula for $\pi^*_B$.

A. The effective loss rate $\pi^*_B$ for an arbitrary schedule

Let $c$ be a $n$-tuple representing a particular failure configuration; $c_i$, $1 \leq i \leq n$, takes the value $G$ (resp., $B$) if $i$th FEC packet is transmitted (resp., lost). By considering all possible failure configurations $c$ we can compute the effective loss rate $\pi^*_B$ for a given schedule $S$ as follows:

$$\pi^*_B = \frac{1}{k} \sum_{c} D(c) \cdot P(c),$$

(3)

where $D(c)$ is the number of lost data packets (after the FEC recovery) for a given failure configuration $c$. For a systematic FEC $(n, k)$ we have

$$D(c) = \left\{ \begin{array}{ll} 0 & \text{if } \sum_{i=1}^{n} 1_{c_i=B} \leq n-k \\ \sum_{i=1}^{k} 1_{c_i=B} & \text{otherwise.} \end{array} \right.$$ 

In order to compute the probability $P(c)$ of a failure configuration $c$, we consider the $R$ paths separately, as follows. Denote by $T(r)$ the vector of length $n_r$ with departure times of packets scheduled by $S$ on path $P_r$. Similarly, let $G(i)$ be the $n_r$-element vector with the failure configuration on path $P_r$ defined by $c$. As the $R$ paths are independent, the probability $P(c)$ is

$$P(c) = \prod_{r=1}^{R} P(c(r)),$$

(4)

where $P(c(r))$ is the probability of a failure configuration $c(r)$ on path $P_r$. The derivation of $P(c(r))$ for the Continuous Time Markov Chain loss model is straightforward. Indeed, denote by $p_{i,j}^{(r)}(\tau)$ the probability of transition from state $i$ to state $j$ on path $P_r$ in time $\tau$, i.e.,

$$p_{i,j}^{(r)}(\tau) = P[X_r(\tau) = j | X_r(0) = i].$$

From classical Markov Chain analysis we have:

$$p_{G,G}^{(r)}(\tau) = \pi^{(r)}_G + \pi^{(r)}_B, \quad \alpha$$

$$p_{G,B}^{(r)}(\tau) = \pi^{(r)}_B - \pi^{(r)}_G, \quad \alpha$$

$$p_{B,G}^{(r)}(\tau) = \pi^{(r)}_G - \pi^{(r)}_G, \quad \alpha$$

$$p_{B,B}^{(r)}(\tau) = \pi^{(r)}_B + \pi^{(r)}_G, \quad \alpha$$

where $\alpha = \exp (- (\mu^{(r)}_G + \mu^{(r)}_B) \tau)$. Now $P(c(r))$ can be easily computed. For example, for $c(r) = GBB$ we have

$$P(c(r) = GBB) = \pi^{(r)}_G \cdot p_{G,B}^{(r)}(\tau_1) \cdot p_{B,B}^{(r)}(\tau_2),$$

where $\tau_i$ is the time interval between the $i$th and $(i+1)$th FEC packet scheduled by $S$ on path $P_r$, i.e, $\tau_i = T_{i+1}^{(r)} - T_{i}^{(r)}$. More generally,

$$P(c^{(r)}) = \pi^{(r)}_{c_{(r)}} \prod_{i=1}^{n-1} p_{c_i^{(r)}, c_{i+1}^{(r)}}(T_{i+1}^{(r)} - T_{i}^{(r)}).$$

(5)

Finally, we plug (6) and (4) to (3), to obtain

$$\pi^*_B = \frac{1}{k} \sum_{c} D(c) \prod_{r=1}^{R} \pi^{(r)}_{c_{(r)}} \prod_{i=1}^{n-1} p_{c_i^{(r)}, c_{i+1}^{(r)}}(T_{i+1}^{(r)} - T_{i}^{(r)}).$$

(7)
concatenation of events, e.g., \( G(\frac{a}{b}) \) (resp., \( \frac{a}{b}B \)) means that any \( b \) out of block of \( a \) consecutive packets are lost and that this block is preceded by a good packet (resp., followed by a bad packet). We can now compute \( \mathbb{P}(F_r = j_r) \) by conditioning on the state of the first packet that conforms the packet loss stationary distribution:

\[
\mathbb{P}(F_r = j_r) = \mathbb{P}(G(\lfloor \frac{n_r}{j_r-1} \rfloor)) + \mathbb{P}(B(\lfloor \frac{n_r}{j_r-1} \rfloor)) =  
\pi_G^r \cdot \mathbb{P}(\lfloor \frac{n_r}{j_r} \rfloor | G) + \pi_B^r \cdot \mathbb{P}(\lfloor \frac{n_r}{j_r-1} \rfloor | B), \tag{9}
\]

where \( \mathbb{P}(\lfloor \frac{a}{b} \rfloor | q), q \in \{G, B\} \), is the probability that any \( b \) out of a consecutive packets are lost given that this block is preceded by a packet in state \( q \). Although no general closed form of \( \mathbb{P}(\lfloor \frac{a}{b} \rfloor | q) \) is known, it can be calculated by the recursive approach first proposed in [17] and extended e.g. in [4,15]. We show in Appendix the details of this computation. It takes \( \pi_B^r, 1/\mu_B^r \) and \( T_r \) as parameters, and directly uses the relations (5) above.

In order to find \( \mathbb{E}[D_r | F_r = j_r] \), we first derive \( \mathbb{E}[D_r = i, F_r = j_r] \). Let us consider the \( k_r \) data packets and the \( n_r - k_r \) redundancy packets separately, and additionally condition on the state of the last data packet as follows.

\[
\mathbb{P}(D_r = i, F_r = j_r) =  
\mathbb{P}(k_r-i|G) \cdot \mathbb{P}(n_r-k_r|G) + \mathbb{P}(k_r-i+1|B) \cdot \mathbb{P}(n_r-k_r|B) =  
\mathbb{P}(G|k_r-i) \cdot \mathbb{P}(\lfloor \frac{n_r-k_r}{j_r-1} \rfloor | G) + \mathbb{P}(B|k_r-i+1) \cdot \mathbb{P}(\lfloor \frac{n_r-k_r}{j_r-1} \rfloor | B) =  
\pi_G^r \cdot \mathbb{P}(\lfloor \frac{k_r-i}{j_r} \rfloor | G) \mathbb{P}(\lfloor \frac{n_r-k_r}{j_r-1} \rfloor | G) + \pi_B^r \mathbb{P}(\lfloor \frac{k_r}{j_r+1} \rfloor | B) \mathbb{P}(\lfloor \frac{n_r-k_r}{j_r-1} \rfloor | B). \tag{10}
\]

The first equality uses the Markov property of the loss model:

\[
\mathbb{P}(D_r = i, F_r = j_r | \text{last data packet is } q) =  
\mathbb{P}(D_r = i | \text{last data packet is } q) \cdot \mathbb{P}(F_r = j_r | \text{last data packet is } q),
\]

where \( q \in \{G, B\} \). Now it is easy to calculate \( \mathbb{E}[D_r | F_r = j_r] \), because

\[
\mathbb{E}[D_r | F_r = j_r] = k_r \sum_{i=0}^{k_r} \frac{\mathbb{P}(D_r = i, F_r = j_r)}{\mathbb{P}(F_r = j_r)}. \tag{11}
\]

We plug (9) and (11) into (8) and obtain a complete formula for the effective loss rate \( \pi_B^r \):
then this boils down to a simple round-robin schedule applied in [4,6,7,9]. In contrast, when the rates differ, a more elaborate approach should be used. For this purpose we adopt the credit-based technique proposed in [10], as follows. Each path is associated with a credit initially equal to 0. Before each FEC packet transmission the credit of every path \( P_r \) is increased by \( n_r/n \). Next, the path with the largest credit is selected to transmit this packet; the credit of this path is decreased by 1. This scheme is iterated until all \( n \) FEC packets are sent.

The Immediate schedule can be interpreted as the following function:

\[
S_{imm} = \text{Immediate}(n_1 \ldots n_R, T)
\]

Two examples of Immediate schedules \( S_{imm} \) are given in Fig. 3 (C) is a single-path schedule, i.e., with \( n_1 = 6 \) and \( n_2 = 0 \), whereas in (D) we use two paths and \( n_1 = n_2 = 3 \).

B. 'Spread' packet scheduling \( S_{spr} \) - our proposal

Under Immediate, all packets are sent as soon as they are generated, i.e., according to (12). Instead, we propose to spread the packets evenly in all the available time on each path. We call this schedule \( S_{spr} = (T_{spr}, R_{spr}) \). Compared with Immediate, Spread additionally takes the path propagation times \( t_1 \ldots t_R \) and the maximal FEC block delay \( t_{spr}^{FEC} \) as parameters, i.e.,

\[
S_{spr} = \text{Spread}(n_1 \ldots n_R, T, t_1 \ldots t_R, t_{spr}^{FEC})
\]

The design of Spread is not straightforward. Indeed, as the \( k \) data packets are generated at the source with spacing \( T \), the paths are inter-dependent, which may easily lead to the violation of the constraint C1. For example, if we schedule packet 1 on \( P_1 \) at time \( T(1) = 0 \) (and \( k > 1 \)), then no other packet on any path can be scheduled before time \( t = T \).

We guarantee the feasibility of Spread as follows. First, we order the paths according to their rates, starting from the path with the highest rate. (When two paths have the same rate, we take the one with a higher path propagation time first.) We consider the paths one by one following this order. For each path \( P_r \) we spread the packets evenly on time interval \( [t(r), t_{spr}^{FEC} - t(r)] \), where \( t(r) \) takes the smallest possible value that satisfies the feasibility condition. (The value of \( t(r) \) usually grows with the number of paths processed.) We iterate this algorithm until all paths have been scheduled.

We present two examples of Spread schedules \( S_{spr} \) in Fig. 3. We use \( t_{spr}^{FEC} = 170ms \) and two different sets of rates: \( n_1 = n_2 = 3 \) in (E) and \( n_1 = 4, n_2 = 2 \) in (F).

Spread is very effective. Indeed, we can prove that

**Theorem 1:** The Spread schedule is optimal for the repetition code \( \text{FEC}(n,1) \).

**Proof of Theorem 1** Under \( \text{FEC}(n,1) \) every data packet is replicated and sent in \( n \) copies; the reception of at least one such copy leads to a success. As there is only one data packet, all the redundancy packets (i.e., the duplicates of the data packet) can be generated already at time \( t = 0 \). This eliminates all the time dependencies between the paths. Therefore, every path \( P_r \) separately must maximize the probability of at least one successful transmission. It is achieved by even spreading...
on the maximal allowed time interval \([0, t_{\text{FEC}}^{\text{opt}} - t_r]\). (The proof for the under repetition code on a single path can be found in [14] ) This, in turn, is exactly what Spread returns under FEC\((n, 1)\).

Spread builds on even packet spreading - a simple and widely accepted guideline that is often thought of as leading to the optimal solution. Indeed, its optimality was proven for some particular cases [14]. But, surprisingly, this is not a general result. Consider for example FEC(4,3) on a single path (i.e., \(R = (1,1,1,1)\)) with loss rate \(\pi_B^{(1)} = 1\%\) and average loss burst length \(1/\mu_B = 5\text{ms}\), and available time interval equal to 15ms. The even spreading schedule \(S_1 = ((0,5,10,15), R)\) yields \(\pi_B = 0.53\%\). But the optimal schedule (found with optimization tools of Mathematica [18]) is \(S_1 = ((0,7,16,12.51,15), R)\) and yields \(\pi_B = 0.50\%\).

This means that Spread does not guarantee optimality in the general FEC\((n, k)\) case. However, we show later in simulations that it usually leads to almost-optimal solutions and is thus an effective and practical rule of thumb.

C. Comparison of \(S_{\text{imm}}\) and \(S_{\text{spr}}\): Optimal schedules \(S_{\text{imm}}^{\text{opt}}\) and \(S_{\text{opt}}^{\text{spr}}\), and loss rate improvement \(\gamma\).

It was shown in previous studies that Immediate multipath is better than a single path communication. The main point we make here is that once we allow for multipath, the Spread schedule \(S_{\text{spr}}^{\text{opt}}\) that we propose in this paper is significantly better than the Immediate schedule \(S_{\text{imm}}\) representing the state of the art.

In order to demonstrate this, we compare the performance of \(S_{\text{imm}}\) and \(S_{\text{spr}}^{\text{opt}}\) in terms of their effective loss rates. What rates \(n_1, \ldots, n_R\) and what FEC block transmission time \(t_{\text{FEC}}\) should we use to make this comparison meaningful and fair? We should allow Immediate and Spread to optimize their rates \(n_1, \ldots, n_R\) independently, given that they impose identical FEC block transmission times \(t_{\text{imm}} = t_{\text{FEC}}^{\text{opt}}\). More precisely, we assume that the FEC parameters \(n\) and \(k\) are fixed, and we proceed in two steps. First, we optimize the rates \(n_1, \ldots, n_R\) of Immediate, such that the effective loss rate \(\pi_B^{\ast}\) is minimized. It results in the optimal Immediate schedule \(S_{\text{imm}}^{\text{opt}}\). This, in turn, specifies \(t_{\text{imm}}^{\text{opt}}\) as shown in (2). In the second step, we set \(t_{\text{FEC}} = t_{\text{imm}}^{\text{opt}}\) and optimize the rates \(n_1, \ldots, n_R\) of Spread, resulting in the optimal Spread schedule \(S_{\text{opt}}^{\text{spr}}\).

Finally, we define the relative effective loss rate improvement \(\gamma\) as the relative gain in \(\pi_B^{\ast}\) due to the usage of optimal Spread instead of optimal Immediate, i.e.,

\[
\gamma = \frac{\pi_B^{\ast}(S_{\text{imm}}^{\text{opt}})}{\pi_B^{\ast}(S_{\text{opt}}^{\text{spr}})}. \tag{13}
\]

The metric \(\gamma\) can be precisely evaluated by formulas (7) and (11). The values of \(\gamma\) can be easily interpreted; for example, \(\gamma > 1\) means that Spread performs better than Immediate.

\[\text{Note that } S_{\text{imm}}^{\text{opt}}\text{ and } S_{\text{opt}}^{\text{spr}}\text{ are optimal subject to their construction constraints presented in IV-A and IV-B respectively.}\]

D. Capacity constraints

So far we have considered the case where every path \(P_r\) can be assigned with any rate \(0 \leq n_r \leq n\). In practice, however, \(P_r\) may have a relatively limited capacity, which would impose a direct constraint on \(n_r\). Fortunately, integrating these constraints in our model is straightforward. Indeed, it is enough to respect them when computing the rates \(n_1, \ldots, n_R\) in \(S_{\text{opt}}^{\text{spr}}\) and \(S_{\text{imm}}^{\text{opt}}\) in IV-C.

V. PERFORMANCE EVALUATION

In this section we evaluate our approach first in simulations and next on real-life traces.

A. Simulation results

The goal of simulations is twofold. First, we verify the correctness of our analytical results. Second, we can test our idea in a fully controlled environment and study the effect of various parameters on the results.

1) Default values of parameters: If not stated otherwise, in our simulations we use the following default values. The data packets are generated at the source with interval \(T = 5\text{ms}\). Next they are encoded by systematic FEC(10, 8) and sent over \(R\) independent paths. For the sake of simplicity we speak mainly of systems with \(R = 2\) paths: \(P_1\) and \(P_2\). It allows us to describe the path propagation time differences by a single parameter \(\Delta t = t_2 - t_1\) that takes the default value \(\Delta t = 100\text{ms}\). Finally, the paths \(P_1\) and \(P_2\) have the same average failure rate \(\pi_1 = 0.01\) and the average loss burst length equal to \(1/\mu_1 = 10\text{ms}\).

2) The effective loss rate \(\pi_B^{\ast}\) as a function of \(\Delta t\): In Fig. 7 we plot the effective loss rate \(\pi_B^{\ast}\) as a function of \(\Delta t\) for four different schedules. Our first observation is that the results obtained in a simulation of the model (circles) fit precisely the analytical curves (plain lines).

Next, we compare the performance of various schedules. As the loss properties of the two paths are identical, the previous techniques described in [4,5,10] split the FEC packets equally between \(P_1\) and \(P_2\). This results in the optimal Immediate schedule \(S_{\text{imm}}^{\text{opt}} = S_{(5,5)}\), i.e., \(n_1 = n_2 = 5\). As this schedule uses multipath transmission, it is not surprising that \(S_{\text{imm}}^{\text{opt}}\) significantly outperforms the single path Immediate schedule \(S_{\text{imm}}^{(10,0)}\). Note also that, by construction, \(\Delta t\) does not affect the performance of any of them.

In contrast, in Spread \(S_{\text{opt}}^{\text{spr}}(5,5)\) we use the same rates as in \(S_{\text{imm}}^{\text{opt}}\), but we spread the packets uniformly within the time budget \(t_{\text{FEC}}^{\text{opt}}\) set by \(S_{\text{opt}}^{(5,5)}\) (similar schedule is shown in Fig. 6). It results in a further decrease of the effective loss rate \(\pi_B^{\ast}\). This difference moderately grows with \(\Delta t\). However, for larger \(\Delta t\) the rates \(5, 5\) become suboptimal under Spread. For instance, in the inset in Fig. 7 we show the performance of Spread under various rate configurations \((n_1, n - n_1)\); the minimum is reached for \((7, 3)\). As described in IV-C allowing for this rate optimization leads to the optimal Spread schedule \(S_{\text{opt}}^{\text{spr}}\). Its advantage over \(S_{\text{imm}}^{(5,5)}\) grows roughly exponentially with \(\Delta t\).
Section IV. The optimal schedule was found with numerical optimization tools. The dashed curve shows the effective loss rate of the optimal schedule, where packets are not necessarily evenly-spaced on each path, as described in a simulation of the model. The size of confidence intervals (not shown) is values according to formula (11), whereas the circles are the results obtained less to exploit and them on the relative loss rate improvement transmission time $\Delta t$. We plot the results in Fig. 9. The gain in FEC block transmission time $t_{\text{FEC}}$ by the usage of Spread instead of Immediate. Parameters: FEC(10,8), $\pi_B^{(1)} = \pi_B^{(2)} = 0.01$, $1/\mu_B^{(1)} = 1/\mu_B^{(2)} = 10\, \text{ms}$, $T = 5\, \text{ms}$. For these parameters, the optimal Immediate rates are $n_1 = n_2 = 5$, which results in the effective loss rate $\pi_B^*(S_{\text{imm}}^{(5,5)}) = 0.24\%$. For Spread we choose the minimal FEC block transmission time $t_{\text{FEC}}^{\text{opt}}$ such that $\pi_B^*(S_{\text{opt}}^{\text{spr}}) \leq \pi_B^*(S_{\text{imm}}^{(5,5)})$.

Finally, we observe that the performance of the optimal Spread schedule $S_{\text{opt}}^{\text{spr}}$ is very close to the global optimum (dashed curve) where packets are not necessarily evenly-spaced, as described in Section IV. This confirms the usefulness of the even-spread guideline that we follow in Spread.

3) Loss rate improvement $\gamma$ as a function of various parameters: Clearly, there are many parameters that affect the performance of the schedules. We study the effect of some of them on the relative loss rate improvement $\gamma$ in Fig. 8.

First, plot (A) confirms that the advantage of Spread over Immediate grows with the path propagation time difference $\Delta t$. Second, with growing packet interval $T$ at the source, the fixed $\Delta t$ becomes a smaller fraction of the entire FEC block transmission time $t_{\text{FEC}}$. As a consequence, there is relatively less to exploit and $\gamma$ drops with $T$, see plot (B). A similar phenomenon can be observed in plot (C), where $t_{\text{FEC}}$ grows due to an increase of the number $n$ of FEC packets.

Finally, in Fig. 8D we vary the loss rate $\pi_B^{(2)}$ of path $P_2$. The difference between path loss rates is a crucial parameter affecting the performance gain of Immediate multipath over the single path transmission. Indeed, if out of two paths one is very lossy and the other one is very good, then the optimal Immediate multipath schedule $S_{\text{imm}}^{\text{opt}}$ uses mainly (or only) the better path, which substantially limits the gain of multipath [5,7]. This is illustrated in plot (D) by the dashed curve; the ratio $\pi_B^*(S_{\text{opt}}^{\text{imm}})/\pi_B^*(S_{\text{opt}}^{\text{spr}})$ of Spread over Immediate. Surprisingly, this is not the case: $\gamma$ remains relatively stable ($3 < \gamma < 6$) for a wide range of values of $\pi_B^{(2)}$. For $\pi_B^{(2)} \approx 0.25$ the path $P_2$ becomes too lossy, and both Immediate and Spread send all packets on $P_1$ only and thus become equivalent.

4) Minimizing $t_{\text{FEC}}$ - decreasing delays and fighting jitter: So far we used Spread to minimize the effective loss rate $\pi_B^*$ while keeping the FEC block transmission time $t_{\text{FEC}}^{\text{spr}}$ not larger than that of Immediate schedule $t_{\text{FEC}}^{\text{imm}}$. Let us now reverse the problem: Let us minimize the FEC block transmission time $t_{\text{FEC}}^{\text{spr}}$ of Spread, while keeping its effective loss rate not larger than that of Immediate, i.e., subject to $\pi_B^*(S_{\text{opt}}^{\text{spr}}) \leq \pi_B^*(S_{\text{imm}}^{\text{opt}})$.

We plot the results in Fig. 9. The gain $t_{\text{FEC}}^{\text{imm}} - t_{\text{FEC}}^{\text{spr}}$ in FEC block transmission time is significant and grows roughly linearly with $\Delta t$, as $t_{\text{FEC}}^{\text{imm}} - t_{\text{FEC}}^{\text{spr}} \approx \Delta t/2$. The reduction of $t_{\text{FEC}}$ brings obvious advantages to delay-constrained applications using the multipath FEC system. First, the effective end-to-end delays get smaller which allows us to reduce the playout time at the destination, keeping the same level of the effective loss rate.

Another important interpretation is related to the delay jitter, i.e., variations of path propagation times. Indeed, in this work, as in most previous works on multipath FEC, we consider the path propagation times constant and focus on (correlated) packet losses only. However, as Spread results in a smaller delay $t_{\text{FEC}}$, it also leaves more space to accommodate potential jitter, naturally making Spread more robust to jitter than Immediate.
Immediate and Spread use the same general FEC parameters. However, in some cases the optimal choice of $n$ and $k$ under Spread may differ from that of Immediate, given the same redundancy $k/n$. For example, according to our additional simulations (not shown here), for the setting in Fig. 7 and $\Delta t = 220ms$, the Spread schedule using FEC(15,12) would outperform Spread with FEC(10,8). Similarly, FEC(12,6) would be better for Spread than FEC(10,5) for $\Delta t > 140ms$. Note, however, that this phenomenon can be observed only for relatively large values of $\Delta t$ that rarely occur in reality.

To conclude, the loss rate improvement $\gamma$ can be manyfold, but its exact value strongly depends on various parameters. First, the advantage of Spread over Immediate grows with path propagation time differences $\Delta t$, but drops with the data packet generation interval $T$ and FEC block size $n$. Second, the optimal Immediate rates are not always optimal under Spread; usually optimal Spread sends more packets on faster paths. Third, although the advantage of Immediate over a single path transmission quickly diminishes with growing differences between the loss rates of the paths, the advantage of Spread over Immediate is relatively stable. Finally, Spread can also achieve FEC block transmission times much smaller than Immediate, still guaranteeing the same or better performance in terms of the effective loss rate. This results not only in smaller effective delays, but also in a higher robustness to delay jitter.

B. Trace-driven PlanetLab evaluation

In the previous section we presented analytical and simulation results where the packet losses were modeled by the Continuous Time Gilbert Model. As any model, it is only an approximation of reality. In this section we feed our simulations with real-life packet loss traces collected in Internet experiments.

1) Data sets: The traces come from two different PlanetLab (PL) [13] experiments. On every path the packets are sent with time-interval $T$, i.e., with the generation rate at the source. Every trace is a sequence composed of symbols $G$ (packet not lost) and $B$ (packet lost).

Every time-constrained experiment on PlanetLab should be designed and interpreted carefully. This is because at any point in time most of PlanetLab nodes are overloaded. Not only their CPU utilization is at 100%, but more importantly the queueing delays experienced by the running processes can be very significant - even up to several seconds between two consecutive accesses to CPU. This results in incorrect propagation time measurements and packet dropping due to incoming buffer overflow at the destination [19,20]. Moreover, the situation changes dynamically. We minimize the effects of these problems by introducing periodic pauses in packet generation and avoiding the highly loaded PlanetLab nodes.

We use the following two data sets.

a) ‘Relays’ - PlanetLab with relays: In this experiment every trace is collected on a two-hop overlay path between three PlanetLab nodes: source, relay and destination. The UDP packets at the source are generated every $T = 5ms$ and sent immediately to the relay that forwards them to the destination. After every one-second-long packet generation period we introduce one second of idle time in order to avoid dropping packets at PlanetLab hosts when the probing traffic is too bursty. We collected more than 5’000 traces, each covering 100 seconds of packet generation time.

In order to further reduce the effect of overloaded PL nodes on the results, for every experiment separately we select the source, relay and destination randomly from 50 currently least loaded PL nodes. As the load estimate we use the number of processes queueing for the CPU and I/O devices; it can be obtained by parsing the file /proc/stat that stores the information about kernel activity.

b) ‘Web sites’ - PlanetLab to popular web sites: This data set consists of 2’839 traces used in [10]. They were col-
lected by sending 16-byte ICMP echo packets from 57 PlanetLab nodes to 55 popular web sites selected from [21]. Next, the ICMP echo-reply packets were captured by Tzqpdump, resulting in traces where packets travel from a PlanetLab node to a web site and back to the original PlanetLab node. The packets were sent every $T = 2ms$. As above, every one-second packet generation time was followed by one-second idle time. Each measurement lasted at least 800 seconds. As in [10], we split it into 40-second long intervals that we call chunks.

Despite the measures we took, in both data sets we find traces with numerous long (100ms and more) blocks of consecutive losses. As this is not caused by network losses, but rather by buffer overflow at the nodes due to CPU queueing, we exclude these traces from our simulations.

2) Simulation setting: In a simulation of a $R$-path scenario we use $R$ traces (one per path) randomly chosen from the pool of all available traces. Thus, by construction, the $R$ traces are independent (typically generated at different times and places in the Internet). For the sake of simplicity, we restrict the presentation to the case $R = 2$.

Our basic metric is loss rate improvement $\gamma$. As described in Section IV-C it optimizes the rates of Immediate and Spread. This optimization is based on the observed traces. One approach to do this is to infer for every path its loss rate $\pi_B^{(r)}$ and the average loss burst length $1/\mu_B^{(r)}$, feed them into the model and optimize the rates as in section V-A. However, this technique has two drawbacks: it introduces errors when measuring the path properties, and assumes a particular packet loss model. We avoid these problems by working directly on the traces - the optimal rates in $S_{opt}^{imm}$ and $S_{opt}^{spr}$ are those that perform best on a given chunk.

We present two types of results. In Oracle we choose the optimal rates for the currently evaluated chunk. In contrast, in Prediction we use the optimal rates of the preceding chunk to evaluate the current chunk. Thus Oracle shows the best achievable results for Immediate and Spread with no prediction errors, whereas Prediction is a practical implementation.

3) Results: In Fig. 10 we present the results for FEC(10, 8). The figure presents the cumulative distribution of the relative loss rate improvement $\gamma$ for $\Delta t = 10ms$ and $\Delta t = 50ms$. We consider the cases where optimal Immediate uses both paths (i.e., $n_1 \neq 0$ and $n_1 \neq 10$) and there is a space for improvement (i.e., $\pi_B^*(S_{opt}^{imm}) > 0$). In about 90% of cases we observe an advantage of Spread over Immediate. For instance, for both data sets under Oracle with $\Delta t = 50ms$, in 50% of cases the loss rate drops by a factor of 3 or more when we use Spread instead of Immediate. For smaller $\Delta t$ the advantage is less pronounced, which is in agreement with the results presented in the previous section.

Surprisingly, in roughly 10% of cases Spread performs slightly worse than Immediate. A possible explanation is that in some traces we can find loss patterns that are periodic, presumably due to other applications running on PlanetLab nodes. If such an unnatural loss pattern gets aligned with the packets scheduled by Spread on one or more paths, then the performance of Spread may drop below Immediate.

Fig. 10. The effective loss rate improvement $\gamma$ (by using Spread instead of Immediate) in trace-driven simulations under FEC(10, 8). We use $R = 2$ independent paths with real-life loss traces; their propagation times differ by $\Delta t$. We consider two data sets: 'Relays' (top) and 'Web sites' (bottom).

Finally, we find our simple prediction method satisfactory, as the Prediction curve is always close to Oracle.

VI. RELATED WORK

The performance of FEC on a single path with correlated loss failures was studied e.g., in [1,14,15]. One common conclusion is that the FEC efficiency drops with growing burstiness of packet losses.

Multipath transmission as a way of de-correlating the packet losses and increasing the performance of FEC was first proposed in [3]. It got more attention recently, e.g., in [4]–[7,9,10]. Multipath was also studied in the context of Multiple Description Coding [11].

In [5] the authors study a multipath FEC system by simulations only, on artificially generated graphs. They also give a heuristic to select from a number of candidate paths a set of highly disjoint paths with relatively small propagation delays.

There are a number of approaches to evaluate analytically the performance of multipath FEC with independent paths and bursty path losses. For instance, [4,6,7] and [10] give four different derivations of the effective loss rate $\pi_B$ (or related metrics) in such a setting. However, in all four cases the resulting formula is only an approximation of the complete solution due to (sometimes very significant) model simplifications. First, [6] [7] use a discrete Gilbert model. Thus two consecutive packets on one path are equally correlated irrespectively of the time intervals between them, which makes the models inherently unable to capture any aspects of varying packet spacing. [10] also uses the a discrete Gilbert model, but adapts the transition matrix appropriately. Second approximation comes when computing the number of lost data packets given that a FEC block cannot be entirely recovered: [4] and
[6] use approximations described at the end of section III-B. [7] simplifies the model by assuming that in such a case all data packets are lost, and [10] assumes that the numbers of lost data packets and redundancy packets are not correlated. Third, [6] considers only a scenario with identical loss statistics on every path. Finally, [10] assumes a large number of active paths \( R \gg 1 \) and small individual path rates \( n_r \ll n \). This allows the authors to apply the central limit theorem and approximate the joint distribution of the number of lost data and redundancy packets by a bivariate normal distribution.

To the best of our knowledge, we are the first ones to give an exact analytical formula for the effective loss rate \( \pi_B^* \) of FEC protection scheme on multiple independent paths with path losses modeled by the Gilbert model. As in most other approaches, we assume that the background cross-traffic is much larger than our own, and thus the load we impose on a path does not affect its loss statistics. Scenarios where this assumption does not hold were studied in [16] in the context of a single path FEC, and in [22] for multipath FEC.

As in [4,9,10,22] we assume the paths to be independent. This can be achieved by detecting correlated paths in end-to-end measurements [23] and treating them as one. Another approach is to find paths that are IP link disjoint, which should be possible if the site is multi-homed. Finally, even if all the available paths are to some extent correlated we can still get some performance benefits [5,6,8,24], though limited [25,26].

Finally and most importantly, to the best of our knowledge no attempt has been made to exploit the path propagation time differences in multipath FEC. Indeed, all the works listed above use some variant of the Immediate schedule, where packets are sent as soon as they arrive at the source. In contrast, in this paper we have proposed the Spread schedule that exploits these propagation time differences and significantly improves the performance.

VII. CONCLUSION

In this paper we started from the observation that the propagation times on multiple paths between a pair of nodes may significantly differ. We proposed to exploit these differences in the context of delay-constrained multipath systems using FEC, by applying the Spread schedule. We have evaluated our solution by a precise analytical approach, and simulations based on both the model and real-life Internet traces. Our studies show that Spread substantially outperforms previous solutions. It achieves a several-fold improvement (reduction) of the effective loss rate. Or conversely, keeping the same level of effective loss rate Spread significantly decreases the FEC block transmission time, which limits the observed delays and helps fighting the delay jitter.

VIII. ACKNOWLEDGEMENTS

We would like to thank the authors of [10] for sharing their loss traces with us, and Patrick Thiran and Dan Jurca for helpful comments. This work is financially supported by grant ManCom 2110 of the Hasler Foundation, Bern, Switzerland.

IX. APPENDIX

A. Recursive equations

Here we derive the probability \( \mathbb{P}(\binom{m}{b} | q) \) that any \( b \) out of \( a \) consecutive packets sent on a path \( P_r \) (with packet interval \( T_r \)) are lost given that this block is preceded by a packet in state \( q \in \{G,B\} \). Although no general closed form of \( \mathbb{P}(\binom{m}{b} | q) \) is known, it can be calculated by the recursive approach first proposed in [17] and extended e.g. in [4,15]. Indeed,

\[
\begin{align*}
\mathbb{P}(\binom{m}{b} | B) &= R(b+1, a+1) \\
\mathbb{P}(\binom{m}{b} | G) &= S(b+1, b-a+1),
\end{align*}
\]

where functions \( R(m,n) \) and \( S(m,n) \) can be calculated as follows [15]:

\[
R(m,n) = \begin{cases} P(n) & \text{for } m=1 \text{ and } n \geq 1 \\ \sum_{i=1}^{n-m+1} p(i)R(m-1,n-i) & \text{for } 2 \leq m \leq n \end{cases}
\]

\[
S(m,n) = \begin{cases} Q(n) & \text{for } m=1 \text{ and } n \geq 1 \\ \sum_{i=1}^{n-m+1} q(i)S(m-1,n-i) & \text{for } 2 \leq m \leq n \end{cases}
\]

where

\[
\begin{align*}
p(i) &= \begin{cases} 1-q & \text{if } i = 1 \\ q(1-p)^{i-2}p & \text{otherwise} \end{cases} \\
P(i) &= \begin{cases} 1 & \text{if } i = 1 \\ q(1-p)^{i-2} & \text{otherwise} \end{cases} \\
q(i) &= \begin{cases} 1-p & \text{if } i = 1 \\ p(1-q)^{i-2} & \text{otherwise} \end{cases} \\
Q(i) &= \begin{cases} 1 & \text{if } i = 1 \\ p(1-q)^{i-2} & \text{otherwise} \end{cases} \\
p &= p_{B,G}(T_r) & \text{given by } (5) \\
q &= p_{B,G}(T_r) & \text{given by } (5)
\end{align*}
\]

B. The effective loss rate \( \pi_B^* \) for non-systematic multipath FEC

All formulas shown so far were derived for the systematic version of FEC. The non-systematic FEC\((n,k)\) is easier to handle, and leads to a simplification of these formulas, as follows.

For an arbitrary schedule the derivation of (7) is the same, except that now the number \( D(c) \) of lost data packets for a given failure configuration \( c \) is

\[
D(c) = \begin{cases} 0 & \text{if } \sum_{i=1}^{n} 1_{c_i=B} \leq n - k \\ k & \text{otherwise} \end{cases}
\]

Consider now the equal spacing on paths. As the number of lost data packets is always \( k \) for at least \( n-k+1 \) lost FEC packets, the formula (11) for the effective loss rate \( \pi_B^* \) gets
simplified to

$$\pi_B^* = \frac{1}{\kappa} \sum_{j=n-k+1}^n k \cdot P(F = j) =$$

$$= \sum_{j=n-k+1}^n \sum_{0 \leq j_1, \ldots, j_R \leq j} P(F_1 = j_1, \ldots, F_R = j_R) =$$

$$= \sum_{j=n-k+1}^n \prod_{r=1}^R P(F_r = j_r) =$$

$$= \sum_{j=n-k+1}^n \sum_{0 \leq j_1, \ldots, j_R \leq j} \prod_{r=1}^R P(F_r = j_r) \cdots \left( \pi_G^{(r)} \cdot P\left( \left\lfloor \frac{n-1}{j_r-1} \right\rfloor \mid G \right) + \pi_B^{(r)} \cdot P\left( \left\lfloor \frac{n-1}{j_r-1} \right\rfloor \mid B \right) \right),$$

where \( P\left( \left\lfloor \frac{n-1}{j_r-1} \right\rfloor \mid G \right) \) or \( P\left( \left\lfloor \frac{n-1}{j_r-1} \right\rfloor \mid B \right) \) are calculated in IX-A

REFERENCES

[1] Wenyu Jiang and Henning Schulzrinne, “Perceived quality of packet audio under bursty losses,” Proc. of Infocom, 2002.
[2] Y. Zhang, N. Duffield, V.Paxson, and S. Shenker, “On the constancy of internet path properties,” ACM SIGCOMM Internet Measurement Workshop, 2001.
[3] N. F. Maxemchuk, “Dispersity routing in store and forward networks,” Ph.D Dissertation, University of Pennsylvania, 1975.
[4] L. Golubchik, J. Lui, T. Tung, A. Chow, W. Lee, G. Franceschini, and C. Anglano, “Multi-path continuous media streaming: what are the benefits?” Performance Evaluation Journal, vol. 39, 2002.
[5] T. Nguyen and A. Zakhor, “Path diversity with forward error correction (pdf) system for packet switched networks,” Proc. of INFOCOM’03, 2003.
[6] X. Yu, J.W. Modestino, and I. V. Bajic, “Modeling and analysis of multipath video transport over lossy networks using packet-level fec,” Proc. of Distributed Multimedia Systems (DMS), 2005.
[7] Evangelos Vergetis, Roch Guérim, and Saswati Sarkar, “Realizing the benefits of user-level channel diversity,” SIGCOMM Comput. Commun. Rev., vol. 35, no. 5, pp. 15–28, 2005.
[8] B. Ribeiro, E. de Souza e Silva, and D. Towsley, “On the efficiency of path diversity for continuous media applications,” Technical Report: UM-CSE-2005-019, 2005.
[9] Hanoch Levy and Haim Zlatokrilov, “The effect of packet dispersion on voice applications in ip networks,” IEEE/ACM Trans. on Netw., vol. 14, no. 2, pp. 277–288, 2006.
[10] Y. Li, Y. Zhang, L. Qiu, and S. Lam, “Smarttunnel: Achieving reliability in the internet,” Proc. of Infocom’07, 2007.
[11] J. Apostolopoulos, “Reliable video communication over lossy packet networks using multiple state encoding and path diversity,” Proc. Visual Communication and Image Processing, VCIP, 2001.
[12] “Dimes,” [http://www.netdimes.org](http://www.netdimes.org)
[13] “Planetlab,” [http://www.planet-lab.org/](http://www.planet-lab.org/)
[14] Jean-Chrysostome Bolot, Sacha Fosse-Parisis, and Don Towsley, “Adaptive fec-based error control for internet telephony,” IEEE Infocom, 1999.
[15] Pascal Frossard, “Fec performances in multimedia streaming,” IEEE Communications Letters, vol. 5, no. 3, pp. 122, 2001.
[16] X. Yu, J.W. Modestino, and X. Tian, “The accuracy of gilbert models in predicting packet-loss statistics for a single-multiplexer network model,” Infocom, 2005.
[17] E. O. Elliott, “A model of the switched telephone network for data communications,” Bell System Technical Journal, vol. 44, no. 1, pp. 89, 1965.
[18] “Mathematica,” [http://www.wolfram.com/](http://www.wolfram.com/)
[19] Yair Amir, Claudiu Danilov, Stuart Goose, David Hedqvist, and Andreas Terzis, “An overlay architecture for high quality voice streams,” IEEE Transactions on Multimedia, vol. 8, no. 6, pp. 1250 – 1262, 2006.