Exploiting the Path Propagation Time Differences in Multipath Transmission with FEC

Maciej Kurant

***Abstract*—We consider a transmission of a delay-sensitive data stream (e.g., a video call) 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 loss rate of such a classic multipath FEC scheme can often be significantly reduced, while keeping the total transmission rate and delay unchanged. Our key observation is that the propagation times on the available paths often significantly differ, usually 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.

***Index Terms*—Multipath transition, FEC, block delay, propagation time, loss rate, total transmission rate, scheduling**

I. Introduction

We consider a transmission of a delay-sensitive data stream from a single source to a single destination. How can we improve the reliability of such a transmission? Traditional ARQ (Automatic Repeat-reQuest) mechanisms impose additional and usually unacceptable delays. 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] - currently the predominant type of losses in the 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 use 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].

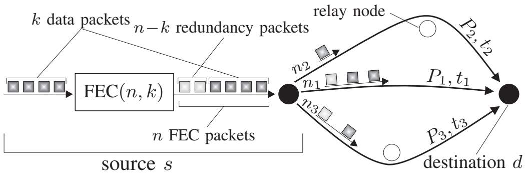


Fig. 1. Illustration of a multipath system with $R = 3$ paths P_1, P_2, P_3 between source s and destination d. t_1, t_2, t_3 are the corresponding path propagation times. k data packets are complimented with $n-k$ redundancy packets, and the resulting n FEC packets are split onto the three paths using the rates n_1, n_2 and n_3, respectively.

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 propose techniques to solve the problems (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

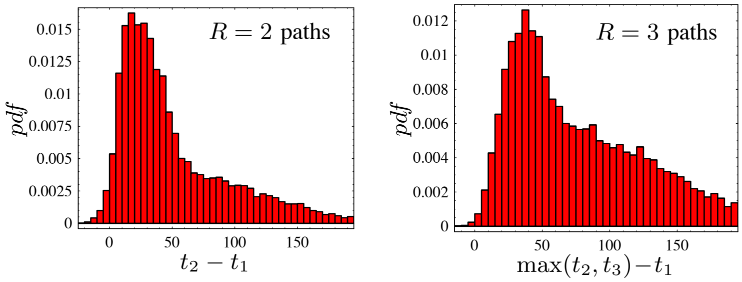
In Fig. 2, we study the path propagation time differences in the real-life Internet. We collected the measurements by running all-to-all traceroutes between 326 nodes in DIMES [12]. These nodes are usually private hosts located at different sites around the world. (We obtained similar results for measurements 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 which we select the $R-1$ paths, according to the procedure given in [5], as follows. 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 minimize the number of overlapping IP links among all R chosen paths (and, again, using the propagation time as a tie-breaker). As a result, we give preference to small-delay paths that are most disjoint with the direct path and between each other.

Manuscript received 29 June 2010; revised 20 December 2010. This is a revised and extended version of a paper presented at *IEEE Infocom'09*. This work was supported by grant ManCom 2110 of the Hasler Foundation, Switzerland, and by SNF grant PBELP2-130871, Switzerland.

M. Kurant is with LCA, EPFL, Lausanne, Switzerland, and EECS Dept, University of California, Irvine (e-mail: maciej.kurant@gmail.com).

Digital Object Identifier 10.1109/JSAC.2011.110512.



		C=1	C=5	C=50
$R = 2$	$t_2 - t_1$ [ms]	43	37	18
$R = 3$	$\max(t_2, t_3) - t_1$	N/A	69	39

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.

According to Fig. 2, for $R = 2$, the best indirect path P_2 has propagation times larger by typically $0 \dots 75ms$ 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 (see table).

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_1 is faster than the slower of the two indirect paths by $\max(t_2, t_3) - t_1 \simeq 0 \dots 150ms$.

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

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. Consider the concrete example in Fig. 3. There exist two paths between the source and the destination, the direct path P_1, and an indirect path P_2 created by employing a relay node. Let $t_1 = 100ms$ and $t_2 = 150ms$ be the propagation delays on P_1 and P_2, respectively. So the path propagation time difference is $\Delta t = 50ms$ (Fig. 3a). We assume that P_1 and P_2 are independent, and have the same loss rate 1% and average loss burst length of $10ms$. The data packets are generated at the source every $T = 5ms$. 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 systematic 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 (the equal rates result from identical loss properties on paths), which decreases π_B^* to 0.148% (d). This solution represents the state of the art in minimizing π_B^*. Note that now the last FEC packet on path P_2 reaches the destination $t_{\text{FEC}} = t_2 + 4 \cdot T = 170$ milliseconds after the generation of the first FEC packet at source. In other words, the application using multipath FEC must accept the (maximal) delay equal to t_{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 an even more significant improvement by sending four packets on P_1 and two packets P_2, i.e., by applying *unequal* sending rates on the paths (f). This results in $\pi_B^* = 0.016\%$, which is almost one order of magnitude smaller than (d). For comparison, we present in (g) the packet-spaced version on a single path.

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 π_B^* may be very significant; here it is 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 (see Fig. 3d vs Fig. 3g). 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 always equal.

C. Organization of this paper

The remainder of this paper is organized as follows. In Section II we fully specify our model, which allows us to precisely state the problem we are solving. Next, in Section III we derive exact analytical expressions for the effective loss rate π_B^* under multipath FEC and an arbitrary schedule. In Section IV 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 V we evaluate our solution analytically, by simulations and by trace-driven simulations fed with real-life Internet traces. In Section VI 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 $t_1, \dots, t_R$, respectively.

A. Path losses

The paths are assumed to be independent. We model bursty losses on each path by the continuous-time version of the

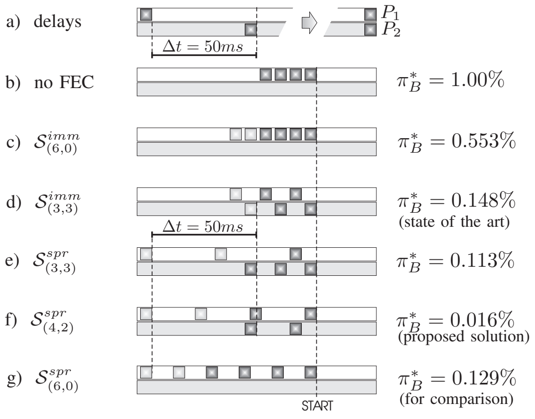


Fig. 3. Illustration of various packet schedules and their performance measured in the effective loss rate π_B^*. We use two independent paths P_1 and P_2 with identical failure distributions. The data packets are generated at the source every $T = 5ms$ and coded with FEC(6,4). **(a)** The path propagation time t_2 on path P_2 is $\Delta t = t_2 - t_1 = 50ms$ larger than on P_1. **(b)** No FEC, single path, the packets are sent at times $0, 5, 10, 15ms$. **(c)** FEC on P_1 only, packets are sent as soon as they are generated, i.e., we use the 'Immediate' schedule $\mathcal{S}^{imm}$. **(d)** Packets alternate between P_1 and P_2 with equal transmission rates $n_1 = n_2 = 3$. 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_1 are maximally spread. **(f)** Packets are split between P_1 and P_2 with optimal rates $n_1 = 4, n_2 = 2$, maximally spread. **(g)** Packets are maximally spread, but on P_1 only (for a comparison purpose).

Gilbert model [4,14]. 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 rth 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: (i) the average loss rate $\pi_B^{(r)}$, and (ii) the average loss burst length $1/\mu_B^{(r)}$. All other parameters can be easily derived from these two, because

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

B. Multipath FEC

We use a *systematic*[1] FEC(n, k) scheme to protect the data packets against losses (see Fig. 1). In a systematic scheme, k unchanged *data* packets are followed by additional $n - k$ *redundancy* packets. As a result, we obtain a FEC block that consists of n *FEC* packets. 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 $F \le n - k$ 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.

TABLE I
BASIC NOTATION USED IN THIS PAPER.

$\mathbb{P}$, $\mathbb{E}$	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	rth 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
$n, k, (n–k)$	the number of FEC, data, and redundancy packets in a FEC block, respectively
n_r	number of FEC packets assigned to P_r (rate of path P_r)
k_r	number of data packets assigned to path P_r
T_r	(constant) spacing of the n_r packets on path P_r
F, D	number of lost FEC and data packets before FEC recovery
π_B^*	effective loss rate, i.e., the expected fraction of lost data packets at the destination after the FEC recovery
t_{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
$\mathcal{S}=(\mathcal{T},\mathcal{R})$	packet scheduling: The ith packet in a FEC block is sent at time $\mathcal{T}(i)$ over path $\mathcal{R}(i)$

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 $\mathcal{S} = (\mathcal{T}, \mathcal{R})$ the schedule of packets in a FEC block, where $\mathcal{T}$ and $\mathcal{R}$ are vectors of length n. The ith FEC packet is sent at time $\mathcal{T}(i)$ over path $\mathcal{R}(i)$, 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_{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 $\mathcal{S}$, t_{FEC} can be easily computed as

$$t_{\text{FEC}} = \max_{1 \le i \le n} \Big(\mathcal{T}(i) + t_{\mathcal{R}(i)}\Big). \qquad (2)$$

For a given schedule, t_{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_{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_{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 $\sum_r k_r = k$.

Moreover, some schedules may make different (usually neighboring) FEC blocks overlap on some paths. This FEC packet interleaving does not affect the scheme and the aggregated path rates. Consequently, our analysis of a single FEC block is also valid for a stream of FEC blocks.

D. Effective loss rate π_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

[1] The non-systematic FEC is easier to handle, but also less efficient. For completeness, we show its analysis in Appendix.

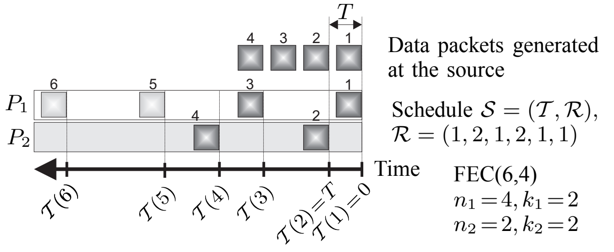


Fig. 4. An illustration of a schedule $\mathcal{S}=(\mathcal{T},\mathcal{R})$ on $R=2$ paths with FEC(6,4). Four data packets numbered 1-4 are generated at the source at equal intervals T; the first one specifies time $t=0$. The $n-k=2$ redundancy packets are numbered 5 and 6. According to the schedule $\mathcal{S}=(\mathcal{T},\mathcal{R})$, the ith FEC packet is sent at time $\mathcal{T}(i)\geq 0$ over path $\mathcal{R}(i)$.

losses observed at the destination, given a maximal value for t_{FEC}. Therefore, we adopt a natural performance metric called *effective loss rate* π_B^*. It is defined as the expected fraction of lost data packets observed at the destination d after an attempt of FEC decoding. Now the problem can be stated as follows:

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

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

III. Derivation of the Effective Loss Rate π_B^*

In order to design a good schedule we must be able to evaluate it. In this section we derive the exact analytical expressions for the effective loss rate π_B^* for a given schedule $\mathcal{S}$.

A. The effective loss rate π_B^* for an arbitrary schedule

First, we derive π_B^* for an *arbitrary* schedule $\mathcal{S}$. 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 ith FEC packet is transmitted (resp., lost). By considering all possible failure configurations c we can compute the effective loss rate π_B^* for a given schedule $\mathcal{S}$ as follows:

$$\pi_B^* \;=\; \frac{1}{k}\sum_{\text{all } c} D(c)\cdot\mathbb{P}(c), \tag{3}$$

where $0\leq D(c)\leq k$ is the number of lost data packets (after the FEC recovery) for a given c. For a systematic FEC(n,k) we have

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

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

$$\mathbb{P}(c)=\prod_{r=1}^{R}\mathbb{P}(c^{(r)}), \tag{4}$$

where $\mathbb{P}(c^{(r)})$ is the probability of a failure configuration $c^{(r)}$ on path P_r. The derivation of $\mathbb{P}(c^{(r)})$ for the Continuous Gilbert 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 τ, i.e.,

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

From the classic Markov Chain analysis we have:

$$\begin{array}{ll} p_{G,G}^{(r)}(\tau)=\pi_G^{(r)}+\pi_B^{(r)}\alpha & p_{G,B}^{(r)}(\tau)=\pi_B^{(r)}-\pi_B^{(r)}\alpha \\ p_{B,G}^{(r)}(\tau)=\pi_G^{(r)}-\pi_G^{(r)}\alpha & p_{B,B}^{(r)}(\tau)=\pi_B^{(r)}+\pi_G^{(r)}\alpha \end{array} \tag{5}$$

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

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

where $\tau_i=\mathcal{T}_{i+1}^{(r)}-\mathcal{T}_i^{(r)}$ is the time interval between the ith and $(i+1)$th FEC packet scheduled by $\mathcal{S}$ on path P_r. Generally,

$$\mathbb{P}(c^{(r)}) \;=\; \pi_{c_1^{(r)}}^{(r)}\prod_{i=1}^{n_r-1}p_{c_i^{(r)},c_{i+1}^{(r)}}^{(r)}(\mathcal{T}_{i+1}^{(r)}-\mathcal{T}_i^{(r)}). \tag{6}$$

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

$$\pi_B^*=\frac{1}{k}\sum_{\text{all } c}D(c)\prod_{r=1}^{R}\pi_{c_1^{(r)}}^{(r)}\prod_{i=1}^{n_r-1}p_{c_i^{(r)},c_{i+1}^{(r)}}^{(r)}(\mathcal{T}_{i+1}^{(r)}-\mathcal{T}_i^{(r)}). \tag{7}$$

B. The effective loss rate π_B^* for even spacing on paths

Equation (7) allows us to compute the effective loss rate π_B^* for any schedule $\mathcal{S}$. However, evaluating (7) is computationally expensive because the main sum is over all the 2^n failure configurations. Thus it can be applied to a relatively small n only. Fortunately, we can significantly reduce the computation complexity by assuming that on each path P_r (separately), the packets are *evenly spaced*, i.e., for all $1\leq i\leq n_r-1$ the intervals $\mathcal{T}_{i+1}^{(r)}-\mathcal{T}_i^{(r)}$ are the same and equal to a constant that we denote by T_r. Indeed, this constraint leads us to a formulation of π_B^* (below) that may take orders of magnitude less time to solve than (7), as shown in Fig. 5.

In order to compute π_B^* under the even-spacing case, we look closer at the packets lost on each path. Denote by F_r and D_r the number of FEC and data packets lost on path P_r, respectively (both before FEC recovery). Now we can rewrite the total number of lost FEC packets as $F=\sum_r F_r$ and the total number of lost data packets as $D=\sum_r D_r$. This decomposition leads us to the following derivation of π_B^*:

$$\begin{aligned}\pi_B^* \;&=\; \frac{1}{k}\sum_{j=n-k+1}^{n}\mathbb{P}(F=j)\cdot\mathbb{E}[D|F=j]= \\ &=\frac{1}{k}\sum_{j=n-k+1}^{n}\sum_{\substack{0\,\leq\, j_1,..,j_R\,\leq\, j\\ j_1+..+j_R=j}}\mathbb{P}(F_1{=}j_1,..,F_R{=}j_R)\cdot\mathbb{E}[D|F_1{=}j_1,..,F_R{=}j_R]= \\ &=\frac{1}{k}\sum_{j=n-k+1}^{n}\sum_{\substack{0\,\leq\, j_1,..,j_R\,\leq\, j\\ j_1+..+j_R=j}}\left(\prod_{r=1}^{R}\mathbb{P}(F_r=j_r)\cdot\sum_{r=1}^{R}\mathbb{E}[D_r|F_r=j_r]\right)\end{aligned} \tag{8}$$

In order to evaluate π_B^*, for each path P_r we need to calculate two components: (i) the probability $\mathbb{P}(F_r=j_r)$

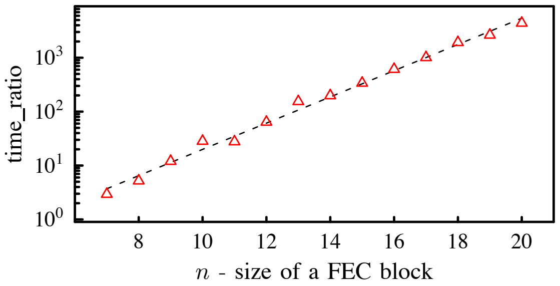


Fig. 5. The time complexity of the effective loss rate π_B^* under an arbitrary schedule (III-A) vs. the even-spaced schedule (III-B): time_ratio is the runtime of Eq. (7) divided by the runtime of Eq. (11). Here we use FEC$(n, 0.7n)$ on two identical paths.

that j_r FEC packets are lost, and (ii) the expected number $\mathbb{E}[D_r|F_r\!=\!j_r]$ of lost data packets given that j_r FEC packets were lost. We achieve this by an approach similar to the one used in [15] in the context of a single path FEC, as follows.

We consider a path P_r and a set of all n_r FEC packets sent on P_r with equal packet interval T_r. We denote by $\genfrac{[}{]}{0pt}{}{a}{b}$ the event that any b out of a consecutive packets are lost. We allow for a concatenation of events, e.g., $G\genfrac{[}{]}{0pt}{}{a}{b}$ (resp., $\genfrac{[}{]}{0pt}{}{a}{b}B$) means that any b out of a 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:

$$\begin{aligned}\mathbb{P}(F_r = j_r) &= \mathbb{P}(G\,\genfrac{[}{]}{0pt}{}{n_r-1}{j_r}) + \mathbb{P}(B\,\genfrac{[}{]}{0pt}{}{n_r-1}{j_r-1}) = \\ &= \pi_G^{(r)}\cdot\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-1}{j_r}\mid G) + \pi_B^{(r)}\cdot\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-1}{j_r-1}\mid B), \quad (9)\end{aligned}$$

where $\mathbb{P}(\genfrac{[}{]}{0pt}{}{a}{b}\ |q)$, $q \in \{G, B\}$, 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 $\mathbb{P}(\genfrac{[}{]}{0pt}{}{a}{b}\ |q)$ is known, it can be calculated by the recursive approach first proposed in [16] and extended e.g. in [15] [4]. We show in the 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 $\mathbb{P}(D_r = i, F_r = j_r)$. 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.

$$\begin{aligned}&\mathbb{P}(D_r = i,\ F_r = j_r) = \\ &= \mathbb{P}(\genfrac{[}{]}{0pt}{}{k_r-1}{i}G)\cdot\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-k_r}{j_r-i}|G)\ +\ \mathbb{P}(\genfrac{[}{]}{0pt}{}{k_r-1}{i-1}\ B)\cdot\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-k_r}{j_r-i}|B)\ = \\ &= \mathbb{P}(G\ \genfrac{[}{]}{0pt}{}{k_r-1}{i})\cdot\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-k_r}{j_r-i}|G)\ +\ \mathbb{P}(B\ \genfrac{[}{]}{0pt}{}{k_r-1}{i-1})\cdot\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-k_r}{j_r-i}|B)\ = \\ &= \pi_G^{(r)}\mathbb{P}(\genfrac{[}{]}{0pt}{}{k_r-1}{i}|G)\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-k_r}{j_r-i}|G)\ +\ \pi_B^{(r)}\mathbb{P}(\genfrac{[}{]}{0pt}{}{k_r-1}{i-1}|B)\mathbb{P}(\genfrac{[}{]}{0pt}{}{n_r-k_r}{j_r-i}|B).\end{aligned}$$

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

$$\begin{aligned}&\mathbb{P}(D_r = i, F_r = j_r \mid \text{last data packet is } q) = \\ &= \mathbb{P}(D_r\!=\!i \mid \text{last data packet is } q)\cdot\mathbb{P}(F_r\!=\!j_r \mid \text{last data packet is } q),\end{aligned}$$

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

$$\mathbb{E}[D_r|F_r = j_r] \ = \ \sum_{i=0}^{k_r} i\cdot\frac{\mathbb{P}(D_r = i, F_r = j_r)}{\mathbb{P}(F_r = j_r)}. \quad (10)$$

We plug (9) and (10) into (8) and obtain a complete formula for the effective loss rate π_B^* (11).

where every term of type $\mathbb{P}(\genfrac{[}{]}{0pt}{}{a}{b}\ |G)$ or $\mathbb{P}(\genfrac{[}{]}{0pt}{}{a}{b}\ |B)$ is calculated through the set of recursive equations given in the Appendix.

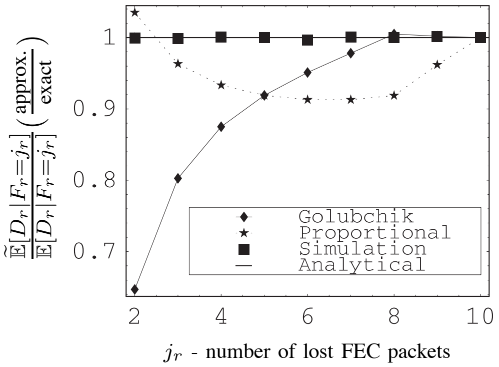


Fig. 6. Approximations of $\mathbb{E}[D_r|F_r = j_r]$ normalized by the correct value given by (10). Here $n_r = 10$, $k_r = 8$, $\pi_B^{(r)} = 0.01$ and $1/\mu_B^{(r)} = 2$.

To the best of our knowledge, Equation (11) is the first exact solution of this model. Indeed, all previous works used some approximations of $\mathbb{E}[D_r|F_r = j_r]$. In [4] the authors approximate $\mathbb{E}[D_r|F_r = j_r]$ by assuming that any configuration of j losses among the n FEC packets is equally likely; we call this approach 'Golubchik'. In [6,23] the authors use an intuitive linear formula, i.e., $\mathbb{E}(D_r|F_r = j_r) = \frac{k_r}{n_r}j_r$. Although not mentioned in the papers this is only an approximation that is exact only when $k_r, n_r \to \infty$; we refer to it as 'Proportional'. We illustrate the differences between these approximations and the real values in Fig. 6.

IV. The Design of the Schedule $\mathcal{S}$

In the previous section, we derive an exact formula for the effective loss rate π_B^* under a given schedule $\mathcal{S}$. Here we focus on the design of a good schedule that results in small π_B^*.

Not all schedules are applicable in practice. Indeed, both (i) the maximal allowed FEC block transmission time t_{FEC} and (ii) the packet interval T at the source impose important scheduling constraints. We say that a schedule is *feasible* if all three of the following conditions are satisfied:

C1 $\mathcal{T}(i) \geq (i-1)\cdot T$ for $1\!\leq\! i\!\leq\! k$, i.e., no data packet is sent before it is generated at the source.

C2 $\mathcal{T}(i) \geq (k-1)\cdot T$ for $k < i \leq n$, i.e., no redundancy packet is sent before all data packets have been generated.

C3 $\mathcal{T}(i) + t_{\mathcal{R}(i)} \leq t_{\text{FEC}}$ for $1\!\leq\! i\!\leq\! n$, i.e., all FEC packets should arrive at the destination before the deadline.

In our discussion below, we first fix the path rates $n_1, \ldots, n_R$ and, given this constraint, we propose two classes of schedules. The first one, called *Immediate* ($\mathcal{S}^{imm}$), reflects the state of the art, whereas the second one, *Spread* ($\mathcal{S}^{spr}$), is our proposal. Next, we allow for independent optimization of rate allocation $n_1, \ldots, n_R$, which leads to optimal versions of these schedules, $\mathcal{S}^{imm}_{opt}$ and $\mathcal{S}^{spr}_{opt}$, respectively.

A. 'Immediate' packet scheduling $\mathcal{S}^{imm}$ - state of the art

The *Immediate* schedule $\mathcal{S}^{imm} = (\mathcal{T}^{imm}, \mathcal{R}^{imm})$ represents the approach used in [4]–[7,9,10]. As the name suggests,

$$\pi^*_B \;=\; \frac{1}{k}\sum_{j=n-k+1}^{n}\;\sum_{\substack{0\le j_1,..,j_R\le j\\ j_1+..+j_R=j}}\left(\prod_{r=1}^{R}\left(\pi_G^{(r)}\cdot\mathbb{P}([{}^{n_r-1}_{j_r}]\,|G)+\;\pi_B^{(r)}\cdot\mathbb{P}([{}^{n_r-1}_{j_r-1}]\,|B)\right)\right)\cdot$$
$$\cdot\left(\sum_{r=1}^{R}\sum_{i=0}^{k_r} i\cdot\frac{\pi_G^{(r)}\cdot\mathbb{P}([{}^{k_r-1}_{i}]\,|G)\cdot\mathbb{P}([{}^{n_r-k_r}_{j_r-i}]\,|G)\;+\;\pi_B^{(r)}\cdot\mathbb{P}([{}^{k_r-1}_{i-1}]\,|B)\cdot\mathbb{P}([{}^{n_r-k_r}_{j_r-i}]\,|B)}{\pi_G^{(r)}\cdot\mathbb{P}([{}^{n_r-1}_{j_r}]\,|G)+\;\pi_B^{(r)}\cdot\mathbb{P}([{}^{n_r-1}_{j_r-1}]\,|B)}\right), \tag{11}$$

Immediate sends the data packets as soon as they are generated, i.e., every time interval T. The redundancy packets use the same spacing T. So in general

$$\mathcal{T}^{imm}(i) = (i-1)\cdot T \quad \text{for } 1\le i\le n. \tag{12}$$

This specifies *when* the FEC packets are sent, but not on which path. A good and commonly used guideline for $\mathcal{R}^{imm}$ is to spread the packets on each path separately with (roughly) even spacing [10]. When the rates are equal, i.e., $n_1{=}n_2{=}\ldots{=}n_R$, 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].

The Immediate schedule can be interpreted as a function

$$\mathcal{S}^{imm} = Immediate(n_1 ... n_R,\ T).$$

Two examples are given in Fig. 3: (c) is a single-path schedule 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 $\mathcal{S}^{spr}$ - our proposal

Under Immediate, all packets are sent as soon as they are generated. We propose, instead, to *spread the packets evenly in all the available times on each path*. We call this schedule *Spread* $\mathcal{S}^{spr} = (\mathcal{T}^{spr}, \mathcal{R}^{spr})$.[2] Compared with Immediate, Spread additionally takes as parameters the path propagation times $t_1 ... t_R$ and the maximal FEC block delay t^{spr}_{FEC}, i.e.,

$$\mathcal{S}^{spr} = Spread(n_1 ... n_R,\ T,\ t_1 ... t_R,\ t^{spr}_{\text{FEC}}).$$

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 $\mathcal{T}(1) = 0$ (and $k > 1$), then no other packet on any path can be scheduled before time $t = T$.

In order to guarantee feasibility, we define Spread as follows. First, we order the paths according to their rates $n_1 ... n_R$, 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 such path P_r, we spread the packets evenly on time interval $[t^{(r)}, t^{spr}_{\text{FEC}} - t_r]$, 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 in Fig. 3. We use $t^{spr}_{\text{FEC}} = 170ms$ and two different sets of rates: $n_1{=}n_2{=}3$ in (e) and $n_1{=}4$, $n_2{=}2$ in (f).

Spread builds on *even packet spreading* - a simple and widely accepted guideline that is often thought of as leading to the optimal solution. Indeed, we can prove the following:

Theorem 1: The Spread schedule is optimal for the repetition code FEC$(n,1)$.

Proof: Under 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., duplicates of the data packet) can be generated already at time $t = 0$. This eliminates the time dependencies between the paths. Therefore, each path P_r must maximize the probability of at least one successful transmission. This probability was proved to be maximized when the n_r packets on P_r are spread *evenly* on the time interval $[0, t^{spr}_{\text{FEC}}{-}t_r]$ (the proof can be found in [14] and holds only for the repetition code). This, in turn, is exactly what Spread returns for every path under FEC$(n,1)$. ■

However, the even packet spreading is *not* always optimal. Consider for example FEC(4,3) on a single path (i.e., $\mathcal{R} = (1,1,1,1)$) with loss rate $\pi^{(1)}_B = 1\%$ and average loss burst length $1/\mu^{(1)}_B = 5ms$, and available time interval equal to 15ms. The even spreading schedule $\mathcal{S}_1 = ((0,5,10,15),\mathcal{R})$ yields $\pi^*_B = 0.53\%$. But the optimal schedule (found with optimization tools of Mathematica) is $\mathcal{S}_1 = ((0,7.16,12.51,15),\mathcal{R})$ 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 close-to-optimal solutions and is thus an effective and practical rule of thumb.

C. Comparison of $\mathcal{S}^{imm}$ and $\mathcal{S}^{spr}$: Optimal schedules $\mathcal{S}^{imm}_{opt}$ and $\mathcal{S}^{spr}_{opt}$, and loss rate improvement γ.

It was shown in previous studies that an Immediate multipath communication is better than a single path communication. The main point we make here is that under multipath, the Spread schedule $\mathcal{S}^{spr}$ that we propose in this paper is significantly better than the Immediate schedule $\mathcal{S}^{imm}$ that represents the state of the art.

In order to demonstrate this, we compare the performance of $\mathcal{S}^{imm}$ and $\mathcal{S}^{spr}$ in terms of their effective loss rates. What rates $n_1 \ldots n_R$ and what FEC block transmission time t_{FEC} should we use to make this comparison meaningful and fair? We should allow Immediate and Spread to optimize independently their rates $n_1 \ldots n_R$, given that they impose

[2] SPREAD can be developed as 'Space Packets Regularly Exploiting Asymmetry in Delays'. AcronymCreator [17] is a great tool that helps creating such meaningful acronyms.

identical FEC block transmission times $t^{imm}_{\text{FEC}} = t^{spr}_{\text{FEC}}$. More precisely, we assume that the FEC parameters n and k are fixed, and we proceed in two steps. First, we optimize the rates $n_1 \dots n_R$ of Immediate, e.g., by evaluating (11) for all possible configurations[3] and selecting the rates that minimize the effective loss rate π^*_B. This results in the optimal Immediate schedule $\mathcal{S}^{imm}_{opt}$, which, in turn, specifies t^{imm}_{FEC} as shown in (2). In the second step, we set $t^{spr}_{\text{FEC}} = t^{imm}_{\text{FEC}}$, optimize the rates $n_1 \dots n_R$ of Spread, and obtain the optimal Spread schedule $\mathcal{S}^{spr}_{opt}$. [4]

Finally, we define the *relative effective loss rate improvement* γ as the relative gain in π^*_B due to the usage of optimal Spread instead of optimal Immediate, i.e.,

$$\gamma \;=\; \frac{\pi^*_B(\mathcal{S}^{imm}_{opt})}{\pi^*_B(\mathcal{S}^{spr}_{opt})}. \tag{13}$$

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

D. Estimating loss model parameters π_B and $1/\mu_B$

Our calculation of the optimal schedules requires as an input the average loss rate π_B and the loss burst length $1/\mu_B$. In practice, we can passively estimate these parameters based on the recent history of our transmission. Indeed, we show in Section V-B2 an implementation called $Prediction$ that uses such recent history and performs close to optimal $Oracle$.

E. Capacity constraints

So far we have considered the case where every path P_r can be assigned with any rate $0 \le n_r \le n$. 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 exclude the unfeasible values of rates $n_1 \dots n_R$ from the computation of $\mathcal{S}^{spr}_{opt}$ and $\mathcal{S}^{spr}_{imm}$ 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, we use the following default values. The data packets are generated at the source with interval $T = 5ms$. Next, they are encoded by systematic FEC$(10, 8)$ and sent over $R = 2$ independent paths: P_1 with $t_1 = 100ms$ and P_2. By default, $t_2 = 200ms$, meaning that the path propagation time difference is $\Delta t = t_2 - t_1 = 100ms$. Finally, the two paths have the same average failure rate $\pi^{(1)}_B = \pi^{(2)}_B = 0.01$ and the average loss burst length equal to $1/\mu^{(1)}_B = 1/\mu^{(2)}_B = 10ms$.

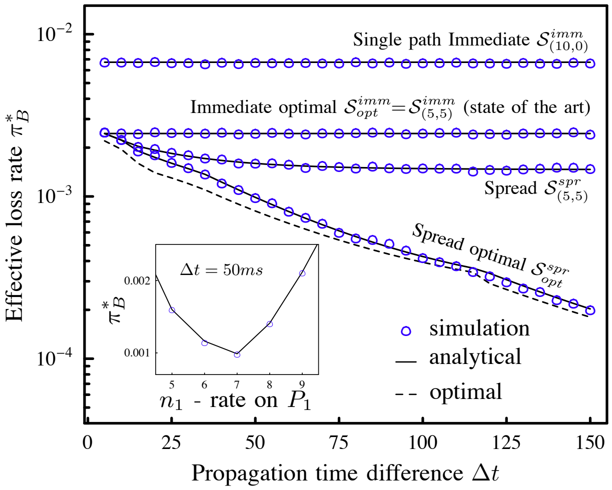


Fig. 7. The effective loss rate π^*_B as a function of path propagation time difference Δt. We use $FEC(10, 8)$ on two independent paths, P_1 with $t_1 = 100ms$ and P_2 with $t_2 = 100ms + \Delta t$, with data packet spacing $T = 5$ at the source. The losses on P_1 and P_2 are modeled by continuous time Gilbert model with the same average failure rate $\pi_B = 0.01$ and the average burst length equal $1/\mu_B = 10ms$. Four schedules are used: • $\mathcal{S}^{imm}_{(10,0)}$ - all packets are sent on a single path P_1 with interval T, • $\mathcal{S}^{imm}_{(5,5)}$ - Immediate with optimal rates $n_1 = n_2 = 5$, • $\mathcal{S}^{spr}_{(5,5)}$ - Spread with $n_1 = n_2 = 5$, • $\mathcal{S}^{spr}_{opt}$ - Spread with the rates n_1, n_2 chosen optimally based on the value of Δt. Additionally, the dashed curve shows the effective loss rate of the *optimal schedule*, where packets are not restricted to even spacing on each path; it was found by numerical optimization tools of Mathematica. **Inset:** π^*_B as a function of rate n_1 on path P_1 for $\Delta t = 50ms$ under Spread. In both figures the plain lines are the theoretical values according to formula (11), whereas the circles are the results obtained in a simulation of the model. The size of confidence intervals (not shown) is similar to the size of the circles.

2) The effective loss rate π^*_B as a function of Δt: In Fig. 7 we plot the effective loss rate π^*_B as a function of Δt for four different schedules. Our first observation is that the simulation results fit precisely the analytical curves. This is expected, because our formulas do not use any approximations.

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

In contrast, in Spread $\mathcal{S}^{spr}_{(5,5)}$ we use the same rates as in $\mathcal{S}^{imm}_{opt}$, but we spread the packets uniformly within the time budget t^{imm}_{FEC} set by $\mathcal{S}^{imm}_{(5,5)}$. It results in a further decrease of the effective loss rate π^*_B. This difference moderately grows with Δt. However, for larger Δt 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 $\mathcal{S}^{spr}_{opt}$. Its advantage over $\mathcal{S}^{imm}_{(5,5)}$ grows roughly exponentially with Δt.

Finally, we observe that the performance of the optimal

[3] This step can be easily speeded-up in practice by exploiting the convexity of the loss rate function, as shown in the inset of Fig. 7.

[4] Note that $\mathcal{S}^{imm}_{opt}$ and $\mathcal{S}^{spr}_{opt}$ are optimal subject to their construction constraints presented in IV-A and IV-B, respectively.

Spread schedule $\mathcal{S}_{opt}^{spr}$ 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 γ 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 γ in Fig. 8.

First, plot (a) confirms that the advantage of Spread over Immediate grows with the propagation time difference Δt.

Second, with growing packet interval T at the source, the fixed Δt becomes a smaller fraction of the entire FEC block transmission time t_{FEC}. As a consequence, there is relatively less to exploit and γ drops with T, see plot (b). A similar phenomenon can be observed in plot (c), where t_{FEC} grows due to an increase in 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 the path loss rates is a crucial parameter affecting the performance gain of the Immediate multipath over the single path transmission. Indeed, if out of two paths one is very lossy and the other is very good, then the optimal Immediate multipath schedule $\mathcal{S}_{opt}^{imm}$ 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 $\pi_B^*(\mathcal{S}_{(10,0)}^{imm})/\pi_B^*(\mathcal{S}_{opt}^{imm})$ is largest when the paths have identical loss properties, and quickly diminishes with growing difference between $\pi_B^{(1)}$ and $\pi_B^{(2)}$.

We could expect a similar diminishing effect for the advantage $\gamma = \pi_B^*(\mathcal{S}_{opt}^{imm})/\pi_B^*(\mathcal{S}_{opt}^{spr})$ of Spread over Immediate. Surprisingly, this is not the case; γ 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_{FEC} - decreasing delays and fighting jitter: So far we used Spread to minimize the effective loss rate π_B^* and keep the FEC block transmission time t_{FEC}^{spr} not larger than that of Immediate schedule t_{FEC}^{imm}. Let us now reverse the problem: Let us minimize the FEC block transmission time t_{FEC}^{spr} of Spread, and keep its effective loss rate not larger than that of Immediate, i.e., subject to $\pi_B^*(\mathcal{S}_{opt}^{spr}) \le \pi_B^*(\mathcal{S}_{opt}^{imm})$.

We plot the results in Fig. 9. The gain $t_{\text{FEC}}^{imm} - t_{\text{FEC}}^{spr}$ in FEC block transmission time is significant and grows roughly linearly with Δt, as $t_{\text{FEC}}^{imm} - t_{\text{FEC}}^{spr} \simeq \Delta t/2$. The reduction of t_{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 and keep 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 we consider the path propagation time constant and focus on (correlated) packet losses. However, as Spread results in a smaller t_{FEC}, it also leaves more space to accommodate potential jitter, thus naturally making Spread more robust to jitter than Immediate.

5) Other FEC parameters n, k: So far we assumed that Immediate and Spread use the same general FEC parameters n and k; only the rates on particular paths could be optimized. 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 Δt that rarely occur in reality.

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 [18,19]. Moreover, the situation changes dynamically. In order to minimize these PlanetLab-specific last-machine issues and measure the delays and losses originating from the network routers, we introduce periodic pauses in packet generation and avoid 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 collected by sending 16-byte ICMP echo packets from 57 PlanetLab hosts to 55 popular web sites selected from [20]. Next,

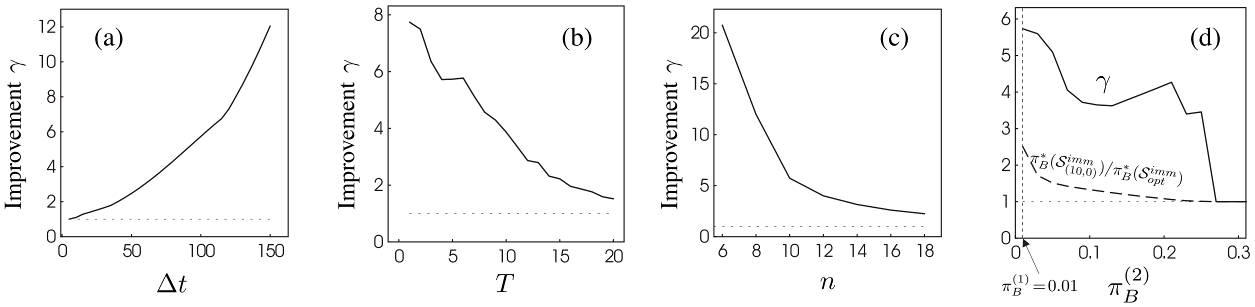


Fig. 8. **(a-d)** Relative loss rate improvement γ due to usage of Spread instead of Immediate as a function of four parameters: **(a)** path propagation time difference Δt, **(b)** packet generation interval T at the source, **(c)** the size n of the FEC block, **(d)** loss rate $\pi_B^{(2)}$ of path P_2. All results (a-d) are analytical, computed for a system with $R = 2$ paths and the following default parameters: FEC(10,8), $\Delta t = 100ms$, $T = 5ms$, $\pi_B^{(r)} = 1\%$, $1/\mu_B^{(r)} = 10ms$, $k = n{-}2$. The **irregular shapes** of the curves are expected, because the computation of γ involves the rates optimization (see IV-C). For instance, in figure (d), going from left to right, the optimal Immediate and Spread rates (n_1, n_2) change gradually (and separately) from $(5, 5)$ to $(10, 0)$; every such rate transition may introduce irregularities in the shape of the curves.

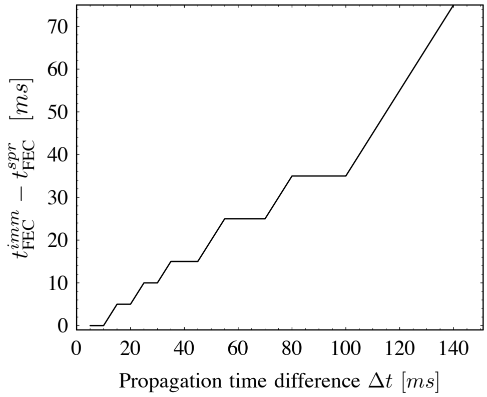


Fig. 9. The gain in FEC block transmission time t_{FEC} by the usage of Spread instead of Immediate. Parameters: FEC(10,8), $\pi_B^{(1)} = \pi_B^{(2)} = 0.01$, $1/\mu_B^{(1)} = 1/\mu_B^{(2)} = 10ms$, $T = 5ms$. For these parameters, the optimal Immediate rates are $n_1 = n_2 = 5$, which results in the effective loss rate $\pi_B^*(\mathcal{S}_{(5,5)}^{imm})=$ 0.24%. For Spread we choose the minimal FEC block transmission time t_{FEC}^{spr} such that $\pi_B^*(\mathcal{S}_{opt}^{spr}) \le \pi_B^*(\mathcal{S}_{(5,5)}^{imm})$.

the ICMP echo-reply packets were captured by Tcpdump, 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 and metrics of interest: 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 γ. 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 $\mathcal{S}_{opt}^{imm}$ and $\mathcal{S}_{opt}^{spr}$ 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 γ 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., $\pi_B^*(\mathcal{S}_{opt}^{imm}) > 0$). 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 Δt 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.

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 is studied e.g., in [1,14,15]. One common conclusion is that the FEC efficiency drops with the increasing burstiness of packet losses.

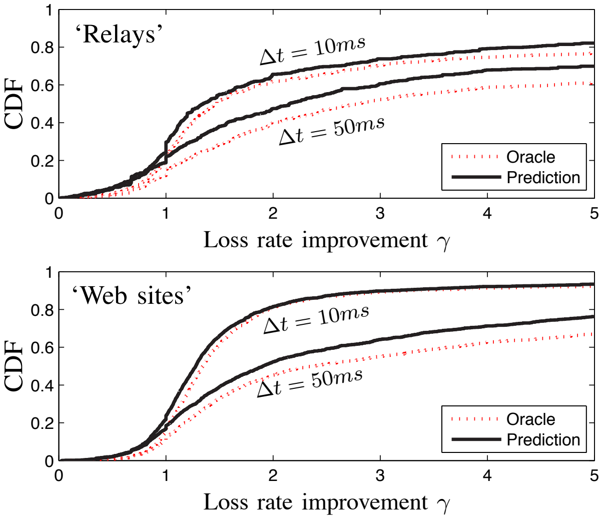


Fig. 10. The effective loss rate improvement γ (by using Spread instead of Immediate) in trace-driven simulations under FEC(10, 8). We use $R = 2$ independent paths with real-life loss traces; their propagation times differ by Δt. We consider two data sets: 'Relays' (top) and 'Web sites' (bottom).

Multipath transmission, as a way of de-correlating the packet losses and increasing the performance of FEC, was first proposed in [3]. It has received more attention recently, e.g., in [4]–[7,9]–[11].

Multipath FEC was also shown to be efficient in combination with TCP, especially in wireless environments [21,22]. However, in this work we focus on delay-sensitive (or time critical) applications such as teleconferencing or gaming, where TCP with its potential retransmissions is not suitable. Instead, we focus on UDP type of transmissions.

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, [4,6,7,10] give four different derivations of the effective loss rate π^*_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 the 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 discrete Gilbert model, but adapts the transition matrix appropriately. The 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 to give an exact analytical formula for the effective loss rate π^*_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 are studied in [23] in the context of a single path FEC, and in [24] for multipath FEC.

As in [4,9,10,24] we assume that the paths are independent. This can be achieved by detecting correlated paths in end-to-end measurements [25] 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 [5,6,8,26], though limited [27,28].

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 with simulations based on both the model and real-life Internet traces. Our studies show that Spread substantially outperforms the previous solutions. It typically achieves a two- to five-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.

Acknowledgements

We would like to thank the authors of [10] for sharing their loss traces with us, and Patrick Thiran, Dan Jurca and Pascal Frossard for helpful comments.

Appendix

A. Recursive equations

Here we derive the probability $\mathbb{P}([^a_b]\,|q)$ 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 $\mathbb{P}([^a_b]\,|q)$ is known, it can be calculated by the recursive approach first proposed in [16] and extended e.g. in [4,15]. Indeed,

$$\begin{aligned}\mathbb{P}([^a_b]\,|B) &= R(b+1, a+1)\\ \mathbb{P}([^a_b]\,|G) &= S(b+1, b-a+1),\end{aligned}$$

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

$$\begin{aligned} p(i) &= \begin{cases} 1-q & \text{if } i=1 \\ q(1-p)^{i-2}p & \text{otherwise} \end{cases} \\ P(i) &= \begin{cases} 1 & \text{if } i=1 \\ q(1-p)^{i-2} & \text{otherwise} \end{cases} \\ q(i) &= \begin{cases} 1-p & \text{if } i=1 \\ p(1-q)^{i-2}q & \text{otherwise} \end{cases} \\ Q(i) &= \begin{cases} 1 & \text{if } i=1 \\ p(1-q)^{i-2} & \text{otherwise} \end{cases} \\ p &= p^{(r)}_{G,B}(T_r) \qquad \text{- given by (5)} \\ q &= p^{(r)}_{B,G}(T_r) \qquad \text{- given by (5)} \end{aligned}$$

B. The effective loss rate 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 π^*_B gets simplified to

$$\begin{aligned} \pi^*_B &= \frac{1}{k} \sum_{j=n-k+1}^{n} k \cdot \mathbb{P}(F=j) = \\ &= \sum_{j=n-k+1}^{n} \sum_{\substack{0 \leq j_1, \ldots, j_R \leq j \\ j_1 + \ldots + j_R = j}} \mathbb{P}(F_1 = j_1, \ldots, F_R = j_R) = \\ &= \sum_{j=n-k+1}^{n} \sum_{\substack{0 \leq j_1, \ldots, j_R \leq j \\ j_1 + \ldots + j_R = j}} \prod_{r=1}^{R} \mathbb{P}(F_r = j_r) = \\ &= \sum_{j=n-k+1}^{n} \sum_{\substack{0 \leq j_1, \ldots, j_R \leq j \\ j_1 + \ldots + j_R = j}} \prod_{r=1}^{R} \cdots \\ &\quad \cdots \Big(\pi^{(r)}_G \cdot \mathbb{P}([{}^{n_r-1}_{j_r}] \mid G) + \pi^{(r)}_B \cdot \mathbb{P}([{}^{n_r-1}_{j_r-1}] \mid B) \Big), \end{aligned}$$

where $\mathbb{P}([{}^{a}_{b}] \,|G)$ or $\mathbb{P}([{}^{a}_{b}] \,|B)$ are calculated in Appendix A.

References

[1] W. Jiang and H. Schulzrinne, "Perceived quality of packet audio under bursty losses," *Proc. 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. Franceschinis, 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. Infocom*, 2003.

[6] X. Yu, J. Modestino, and I. V. Bajic, "Modeling and analysis of multipath video transport over lossy networks using packet-level fec," *Proc. Distributed Multimedia Systems (DMS)*, 2005.

[7] E. Vergetis, R. Guérin, and S. 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-CS-2005-019*, 2005.

[9] H. Levy and H. Zlatokrilov, "The effect of packet dispersion on voice applications in ip networks," *IEEE/ACM Trans. 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. 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*.

[13] "Planetlab," *http://www.planet-lab.org/*.

[14] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive fec-based error control for internet telephony," *IEEE Infocom*, 1999.

[15] P. Frossard, "Fec performances in multimedia streaming," *IEEE Communications Letters*, vol. 5, no. 3, p. 122, 2001.

[16] E. O. Elliott, "A model of the switched telephone network for data communications," *Bell System Technical Journal*, vol. 44, no. 1, 1965.

[17] "Acronym creator," *http://AcronymCreator.net*.

[18] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis, "An overlay architecture for high quality voip streams," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1250 – 1262, 2006.

[19] J. Sommers and P. Barford, "An active measurement system for shared environments," *Internet Measurement Conference*, 2007.

[20] "100hotsites," *http://www.100hotsites.com*.

[21] V. Sharma, S. Kalyanaraman, K. Kar, K.K. Ramakrishnan, and V. Subramanian, "MPLOT: A Transport Protocol Exploiting Multipath Diversity Using Erasure Codes," *Infocom*, 2008.

[22] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath live streaming via TCP: Scheme, performance and benefits," *ACM Trans. Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 5, no. 3, 2009.

[23] X. Yu, J. Modestino, and X. Tian, "The accuracy of gilbert models in predicting packet-loss statistics for a single-multiplexer network model," *Infocom*, 2005.

[24] A. L. H. Chow, L. Golubchik, J. C. S. Lui, and W.-J. Lee, "Multi-path streaming: optimization of load distribution," *Perform. Eval.*, vol. 62, no. 1-4, pp. 417–438, 2005.

[25] D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 381–395, 2002.

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

[27] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," *Proc. IMC'03*, 2003.

[28] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Exploring the performance benefits of end-to-end path switching," *in IEEE ICNP*, 2004.

Maciej Kurant received a M.Sc. degree from Gdansk University of Technology, Poland, in 2002, and a Ph.D. degree from EPFL, Lausanne, Switzerland, in 2009. Currently, he is a postdoc at University of California, Irvine.

His main areas of research interest include sampling and inference from large-scale networks (such as Internet topologies or the human brain), multipath routing with FEC, and survivability in WDM networks. He is also a founder of AcronymCreator.net - a tool that helps creating new, meaningful acronyms.