Multi-path Continuous Media Streaming: What are the Benefits?

L. Golubchik[*] J. C.S. Lui[†] T. F. Tung[†] A. L.H. Chow[†] W.-J. Lee [‡]
G. Franceschinis[§] C. Anglano[§]

Abstract

Quality of service (QoS) in delivery of continuous media over the Internet is still relatively poor and inconsistent. Although many such applications can tolerate some degree of missing information, significant losses degrade an application's QoS. One approach to providing QoS for continuous media applications over the Internet is to use the IntServ model for signaling (e.g., RSVP) and resource reservation in all routers along the streaming path. However, this approach suffers from scalability and deployment problems. In contrast, in this paper we investigate the potential benefits of mitigating the QoS guarantee problem through the exploitation of multiple paths existing in the network between a set of senders and a receiver of continuous media. One advantage of this approach is that the complexity of QoS provision can be pushed to the network edge and hence improve the scalability and deployment characteristics while at the same time provide a certain level of QoS guarantees.

Our focus in this work is on providing a fundamental understanding of the benefits of using multiple paths to deliver continuous media over best-effort wide-area networks. Specifically, we consider pre-recorded continuous media applications (as in video-on-demand systems) and use the following metrics in evaluating the performance of multi-path streaming as compared to single-path streaming: (a) data loss rate, (b) conditional error burst length distribution, and (c) lag1-autocorrelation. The results of this work can be used in guiding the design of multi-path continuous media systems streaming data over best-effort wide-area networks.

1 Introduction

Quality of service (QoS) in streaming of continuous media over the Internet is still poor and inconsistent. The degradation in quality of continuous media applications, involving delivery of video and audio, is partly due to variations in delays as well as losses experienced by packets sent through wide-area networks. Although many such applications can tolerate some degree of missing information, significant losses degrade an application's quality of service. One approach to providing QoS for continuous media applications over the Internet is to use the IntServ model for signaling (e.g., RSVP) and resource reservation in all routers along the streaming path. However, this approach suffers from scalability and deployment problems. In contrast, in this work we investigate the potential benefits of providing QoS guarantees in continuous media delivery through

[*]Computer Science Department, University of Southern California, Los Angeles, CA 90089, leana@cs.usc.edu. This work was partly done while the author was with the University of Maryland.

[†]Computer Science & Engineering Dept, The Chinese University of Hong Kong.

[‡]Department of Computer Science, University of Maryland at College Park.

[§]Dipartimento di Scienze e Tecnologie Avanzate, Università del Piemonte Orientale.

the *exploitation of multiple paths* existing in the network between a set of senders and a receiver. One advantage of this approach is that the complexity of QoS provision can be pushed to the network edge (an original design principle of the Internet) and hence improve the scalability and deployment characteristics while at the same time provide a certain level of QoS guarantees. Our focus in this work is on providing a fundamental understanding of the benefits of using *multiple paths* to deliver continuous media data (such as video) destined for a particular receiver, i.e., this data is fragmented into packets and the different packets take alternate routes to the receiver.

There are a number of approaches to accomplishing a multi-path data delivery, and we describe the specific approach considered in our system below. We first note that such paths do not have to be completely disjoint, i.e., it is sufficient for them to have disjoint points of congestion or bottlenecks. Existence of multiple paths with *disjoint bottlenecks* includes the following potential benefits.

- *Reduction in correlation between consecutive packet losses.* Although a continuous media (CM) application can tolerate some missing information, a large number of consecutive packet losses not only contributes to significant degradation in CM quality but also diminishes ability to correct such losses through error correction techniques, e.g., erasure codes. As we will show in this paper, sending data through multiple paths can potentially reduce burst lengths and correlations between consecutive losses and thus improve the quality of delivered data.

- *Increased throughput.* In delivery of continuous media one can tradeoff the quality of the data with the amount of compression achieved, i.e., one can reduce the amount of bandwidth needed to deliver the data at the cost of its quality. Sending data through multiple paths potentially increases the amount of (aggregate) bandwidth available to the application and hence increases the quality of delivered data.

- *Ability to adjust to variations in congestion patterns on different parts of the network.* CM applications are often long lasting (e.g., delivery of a movie might take on the order of hours). Hence, it is reasonable to expect that network conditions will change throughout the delivery of data to a CM application. Since not all paths, in general, would experience the same traffic patterns and congestion, sending data through multiple paths potentially improves the ability to adapt to changes in network conditions.

In general, the use of multiple paths in designing of distributed (over best-effort wide-area networks) continuous media applications requires consideration of the following issues.

- *Determining bottlenecks, joint points of congestion, and network characteristics in general.* To gain the benefits of multi-path streaming described above, one must first determine the paths to be used in delivery of the data. Since it is reasonable to characterize a path using its bottleneck link [2], what we need to be able to do is determine whether a number of paths share points of congestion, i.e., have joint or disjoint bottlenecks [7, 18]. Although this is not necessary in our approach, other approaches to multi-path streaming might require fairly accurate estimation of various network characteristics (refer to Section 5). These are non-trivial problems which are outside the scope of this paper. However, we note that currently we use [18] in our system for detecting shared points of congestion.

- *Effects of redundancy and error erasure schemes.* Some amount of lost data can be reconstructed in CM applications through the use of redundant information, e.g., as in FEC [1] techniques. Hence, in constructing multipath streaming techniques one should take into consideration the effect of redundant information on the final quality of the data and how the erasure codes interact with multi-path delivery.

- *Adaptation schemes under changes in network conditions.* When network conditions change, one can improve the quality of CM by adapting how the data is streamed on multiple paths (e.g., by sending less data on congested paths).

- *Data placement.* Proper placement of data on the servers is an issue in the context of CM applications delivering *pre-stored* data, for instance, a video-on-demand application (in contrast to a video conferencing application where data is produced "live"). Inappropriate data placement can adversely effect servers' performance. For instance, this can occur due to load imbalance problems arising from the fact that only specific parts of the data are being delivered from a particular server as well as the fact that specific data required might change over the course of the application, as the system adapts to congestion patterns in the network. This in turn reduces the quality of service experienced by the CM application (in this case due to server rather than network performance). We note that these problems can be more severe when adaptation schemes (as mentioned above) are used.

- *Data dispersion.* Given that one cannot necessarily rely on the network layer to provide multipath routing, another consideration is how to accomplish the dispersion of data over multiple paths existing in the network between a sender and a receiver of data. This may be an especially important consideration for applications where data is generated live, e.g., a video conferencing application, in contrast to applications where data is pre-recorded (and hence can, for instance, be dispersed to a set of distributed servers in advance of actual data streaming).

- *Need for protocol/network support.* Lastly, some mechanisms for streaming application data over multiple paths might require support from lower layers, such as the network layer. Of course, in this case, ease of deployment is an issue.

Although all these issues are of importance, in this paper we narrow the scope by focusing on:

- delivery of *pre-stored* video, e.g., as in video-on-demand applications (in contrast to delivery of "live" data as in video-conferencing applications);

- *application-level* schemes (which are deployable today over the current Internet) — that is, we assume the use of best-effort IP-based networks, where a specific path is used between any pair of hosts (sender and receiver) on the network **and** this path is determined by a network-level routing algorithm; furthermore, our system does not require specific knowledge of the paths, only the ability to determine whether two paths share a point of congestion, e.g., using [18];

- accomplishment of multiple paths to the same receiver by *distributing servers* across wide-area networks and streaming data from multiple senders simultaneously;

- streaming over the network issues *only* (rather than, e.g., considering server-related problems such as the load balancing issues mentioned above); that is, for the purposes of this paper we assume that the data is fully replicated at all servers and hence any server can deliver any fraction of the CM data.

Our system is depicted in Figure 1, where any server can send any fraction of the continuous media data. More specifically, server i sends fraction α_i of the data expected by the receiver, where

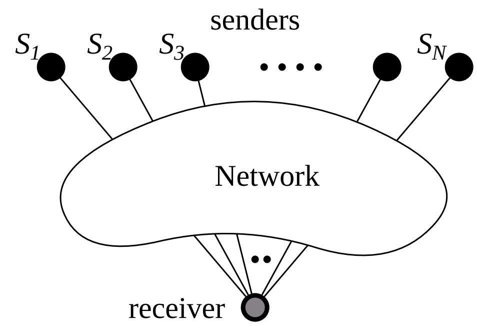


Figure 1: Continuous media system using multipath streaming.

$0 \leq \alpha_i \leq 1$ and $\sum_i \alpha_i = 1$. In general, we assume that the setting and possible adaptation of these fractions (as the delivery of data progresses) is done by the receiver (based on its perceived quality of data and determination of joint points of congestion). The receiver assembles the data from multiple senders and plays it in the appropriate order.

In the remainder of the paper, our focus is on providing the fundamental understanding and on *characterizing the benefits* of the multi-path approach to streaming of pre-stored continuous media data over wide-area networks, under the setup described above. More specifically, we focus on loss characteristics as they are an indication of the resulting quality of the delivered data stream. We believe that the understanding of loss characteristics under a multi-path approach is non-trivial and deserves further attention. We also believe that the work presented here is a step in the right direction. Specifically, the contributions of this paper are as follows. Firstly, we give an analytical characterization of when a multi-path approach is beneficial, as compared to a single path approach, using the following metrics (a) packet loss rate, (b) lag-1 autocorrelation of packet losses, and (c) burst length distribution. (These metrics are defined more formally in Section 2). We also extend this analysis to information loss rate, i.e., we consider the resulting losses after an application of an erasure code. Secondly, we extend the evaluation of the multi-path approach benefits using simulations of the analytical model as well as through more detailed simulations using a packet-level network simulator [8]. These are also performed with and without the use of an erasure code. Our results indicate that: (1) in general, multi-path streaming exhibits better loss characteristics than single-path streaming, (2) use of an erasure code may not necessarily improve data loss characteristics in the case of single-path streaming, while multi-path streaming (with or without use of an erasure code) can improve data loss characteristics, and (3) lag1-autocorrelation of multi-path streaming is usually closer to zero than that of single path streaming, and we believe that this will also result in a higher viewing quality of the received continuous media.

The remainder of this paper is organized as follows. In Section 2 we present our analytical evaluation of the multipath approach described above. This evaluation is extended through simulation, using both the analytical model and a network simulator, in Sections 3 and 4. Section 5 briefly describes some additional considerations in the use of multi-path streaming as well as presents related work on this topic. Our concluding remarks are given in Section 6.

2 Analytical Evaluation

In this section, we present our analysis of the single-path and the multi-path streaming approaches. As mentioned earlier, our main focus is on loss characteristics. We first consider these approaches without the use of erasure codes, so as to understand the basic differences between single and multi-path streaming. We then also consider the changes in loss characteristics when and erasure code, and hence redundant information, is added, as this is another approach to dealing with packet losses. Specifically, we consider a variation of such codes, which we refer to as FEC, as defined below. As in [2], we use a two-state Markov chain, known as the Gilbert model, as our model of a path; as in [2] we characterize the path by its bottleneck link. This model, which is defined more formally below, allows for dependence in consecutive packet losses and should be a more accurate representation of the network than an independent loss model.

We use the following performance measures to quantify the merits of the different streaming approaches (these are defined more formally below):

1. mean data packets loss rate (with and without FEC),
2. conditional burst length distribution, conditioned on there being at least one error (with and without FEC),
3. lag-1 auto-correlation (with and without FEC).

The first performance measure is an obvious approach to comparing single and multi-path streaming (when losses, rather than throughput, are of importance). The other two performance measures are less obvious; however, we believe that they can significantly affect the quality of the viewed continuous medai. To illustrate this point, in the next section we briefly consider a "quality of viewed data" type measure. In subsequent sections we return to the analysis of the streaming techniques.

2.1 Visual Quality of Data

We first give a brief motivation for considering above given performance metrics, and specifically, for considering burst lengths and correlations between losses. We discuss this in the context of video data. Ideally, one would like to have a measure of the quality of the viewed video, as a function of loss characteristics. To the best of our knowledge, there is no such widely accepted measure, and often the quality of a video is evaluated using human observers. However, some metrics have been used in the past, for instance, signal to noise ratio of the resulting video [6]. Hence, we illustrate the effects of bursty losses on the quality of the resulting video (and specifically on the signal to noise ratio) using the following experiment.

Experiment (Effect of Correlated Bursty Losses on Video Quality) : In this experiment, we drop 2% of the frames from video $\mathcal{V}$. These 2% losses are introduced in a variety of "patterns", e.g., the dropped frames can be evenly spaced throughout video $\mathcal{V}$, or they can be more bursty. The details of which frames are dropped, given a particular drop pattern as identified by the burst length, are given in the first two columns of Table 1. Moreover, in evaluating the quality of the resulting video $\mathcal{V}$, we use a common error concealment scheme to make up for a dropped frame. Specifically, a dropped frame is replaced by the previous frame which is successfully received. For example, frame i replaces frames $i+1, i+2, \cdots, i+k$ if frame i is received successfully and frames $i+1, \cdots, i+k$ are lost.

Error Burst Length	Lost Frames Numbers	PSNR (dB)
1	25+k*50 where k$\in \{0, 1, \cdots, 29\}$	39.107 dB
2	{50,51} + k*100 where k$\in \{0, 1, \cdots, 14\}$	38.015 dB
3	{74,75,76} +k*150 where k$\in \{0, 1, \cdots, 9\}$	31.325 dB
5	{123,124,125,126,127} +k*200 where k$\in \{0, 1, \cdots, 5\}$	30.433 dB
15	{368,369,...,381,382} +k*750 where k$\in \{0, 1\}$	28.407 dB
30	{736,737,...,764,765}	29.942 dB

Table 1: Peak signal-to-noise ratio (PSNR) for various bursty loss patterns.

For each possible frame loss pattern, we measure the quality of the received video by computing the peak signal-to-noise ratio (PSNR) as follows. (Note that, a larger value of PSNR implies a higher quality of the video.) In general, for a video of l frames where each frame consists of $m \times n$ pixels, (each containing an RGB value[1] with each of the three colors represented by 8-bits), the PSNR is calculated using the following expression (in dB):

$$SNR_{peak} = 10 \times \log_{10} \frac{255^2}{\left(\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} \sum_{c=1}^{3} (P_1(i,j,k,c) - P_2(i,j,k,c))^2}{3 \times m \times n \times l} \right)}.$$

where $P_s(i, j, k, c)$ is the pixel value at coordinate (i, j) of k-th video frame (of stream s, $s = 1, 2$) and color channel c where $c = 1, 2, 3$, for red, green, and blue, respectively. In our experiment, the values of m,n, and l are 352, 240 and 1500, respectively. The source video in this experiment is using MPEG-1 NTSC settings [5] where each frame is 352×240 (with 29.97 frames per second), hence the values of m and n above. Also, we use approximately the first 50 seconds of this video for this experiment, hence the value of l above. Values for P_1 are obtained from the frame sequence resulting after the drop-and-conceal process while values for P_2 are obtained from the original video frames of $\mathcal{V}$.

Table 1 gives the PSNR values for the different burst patterns. We can observe that given the same amount of information loss (e.g., 2% in our experiment), the PSNR metric can be significantly lower for the more bursty loss patterns, and hence is the quality of the video. Thus, we believe that burst length distribution and correlations between losses are the right metrics for evaluating the goodness of a streaming approach as they directly reflect on the quality of received video.

2.2 Model

Let us now state the path model used in this paper. As in [2], we use a stationary continuous time Gilbert model to characterize the potential correlations between consecutive losses on a path. Under a stationary continuous time Gilbert model, the packet loss process along path k is described by a two state continuous time Markov chain $\{X_k(t)\}$ where $X_k(t) \in \{0, 1\}$. If a packet is transmitted at time t when the state of path k is $X_k(t) = 0$, then no packet loss occurs. On the other hand, the transmitted packet is considered lost if $X_k(t) = 1$. The infinitesimal generator for this Gilbert model of path k is:

$$\boldsymbol{Q}_k = \begin{bmatrix} -\mu_0(k) & \mu_0(k) \\ \mu_1(k) & -\mu_1(k) \end{bmatrix}.$$

[1]Information about the three colors, red, green, and blue.

The stationary distribution of this Gilbert model is $\boldsymbol{\pi}(k) = [\pi_0(k), \pi_1(k)]$ where $\pi_0(k) = \mu_1(k)/(\mu_0(k)+\mu_1(k))$ and $\pi_1(k) = \mu_0(k)/(\mu_0(k) + \mu_1(k))$. Let $p_{i,j}^{(k)}(\tau)$ be the probability that path k is in state j at time $t + \tau$, given that it was in state i at time t, i.e., $p_{i,j}^{(k)}(\tau) = P(X_k(t + \tau) = j|X_k(t) = i)$. From [14], we have that

$$p_{i,j}^{(k)}(\tau) = \begin{cases} \frac{\mu_1(k)}{\mu_0(k)+\mu_1(k)}\left(1 - e^{-[\mu_0(k)+\mu_1(k)]\tau}\right) & i = 1, j = 0, \\ \frac{\mu_0(k)}{\mu_0(k)+\mu_1(k)}\left(1 - e^{-[\mu_0(k)+\mu_1(k)]\tau}\right) & i = 0, j = 1, \\ \frac{\mu_0(k)+\mu_1(k)e^{-(\mu_0(k)+\mu_1(k))\tau}}{\mu_0(k)+\mu_1(k)} & i = 1, j = 1, \\ \frac{\mu_1(k)+\mu_1(k)e^{-(\mu_0(k)+\mu_1(k))\tau}}{\mu_0(k)+\mu_1(k)} & i = 0, j = 0 \end{cases} \tag{1}$$

for all $\tau > 0$.

Throughout the paper we refer to single path streaming as SP streaming and multipath streaming with N paths as MP streaming. Without loss of generality, when paths are homogeneous, we assume that SP streaming always transmits data along path 1. In the evaluation of MP streaming, we assume that the multiple paths have disjoint bottlenecks (or points of congestion) and hence the Gilbert models representing them are independent. Note that, since we represent a path by its bottleneck link, multiple paths with joint points of congestion could just be represented by a single Gilbert model. Lastly, note that our focus is on a streaming application which generates packets at a constant rate; hence our derivations below are done under this assumption.

2.3 Performance Analysis of SP vs. Multi-path Streaming (without FEC)

Let us first derive the average packet loss rate. Unless stated otherwise, below we consider a special case of multi-path streaming, namely dual path, round robin (DPRR) streaming. There are a number of different approaches to distributing data along the multiple paths; here we consider a simple case, i.e., DPRR, wherein each path carries half the application's traffic and the packet transmission is carried out in a round robin manner. That is, odd numbered packets are transmitted along path 1 while even numbered packets are transmitted along path 2. We use this simple scheme for dual path streaming to illustrate the basic performance differences between single and multi-path streaming, so as to gain some basic understanding.

If we assume that the streaming rate does not affect the channel loss characteristics (i.e., the parameters of the Gilbert model), then for the SP case, the average packet loss rate is simply

$$P_{sp}[\text{loss packet}] = \pi_1(1) = \frac{\mu_0(1)}{\mu_0(1) + \mu_1(1)}. \tag{2}$$

For the MP case, assume that we have $N \geq 1$ paths and let α_i be the fraction of the application's workload that is sent along path i where $\sum_{i=1}^{N} \alpha_i = 1$. Then the average packet loss rate for the MP case is

$$P_{mp}[\text{loss packet}] = \sum_{i=1}^{N} \alpha_i \pi_1(i) = \sum_{i=1}^{N} \alpha_i \left(\frac{\mu_0(i)}{\mu_0(i) + \mu_1(i)}\right).$$

If these N paths are homogeneous, then we can simplify the above expression to

$$P_{mp}[\text{loss packet}] = \frac{\mu_0(1)}{\mu_0(1) + \mu_1(1)}. \tag{3}$$

Remark: the implication of Equations (2) and (3) is that if the application's sending rate does not affect the loss characteristics of the path then splitting the data between multiple homogeneous paths does *not* reduce the average packet loss rate, as compared to a single path with the same loss characteristics.

On the other hand, if the application's sending rate can affect the loss characteristics of the path (e.g., sending data with a higher bandwidth may increase the losses), then the average loss rate of the MP approach can be different from that of the SP approach. To illustrate this effect, let λ be the application's mean sending rate and

$$\mu_0(i) = \mathcal{F}(\lambda) \tag{4}$$

$$\mu_1(i) = \mathcal{B}(\lambda) \tag{5}$$

where $\mathcal{F}$ ($\mathcal{B}$) is a continuous non-decreasing (non-increasing) function of λ. Then, we have the following result.

Theorem 1 *If the parameters of the Gilbert model are specified by functions $\mathcal{F}$ and $\mathcal{B}$, then the average packet loss rate under the single path streaming approach will be greater than or equal to the average packet loss rate under the multi-path streaming approach wherein these paths have the same Gilbert's parameters.*

Proof: It is easy to show that the rate of change of the MP average packet loss rate under the homogeneous Gilbert model is:

$$\begin{aligned}
\frac{dP_{mp}[\text{loss packet}]}{d\lambda} &= \frac{d}{d\lambda}\left[\frac{\mathcal{F}(\lambda)}{\mathcal{F}(\lambda)+\mathcal{B}(\lambda)}\right] \\
&= \frac{[\mathcal{F}(\lambda)+\mathcal{B}(\lambda)]\mathcal{F}'(\lambda) - \mathcal{F}(\lambda)[\mathcal{F}'(\lambda)+\mathcal{B}'(\lambda)]}{[\mathcal{F}(\lambda)+\mathcal{B}(\lambda)]^2} \\
&= \frac{\mathcal{B}(\lambda)\mathcal{F}'(\lambda) - \mathcal{F}(\lambda)\mathcal{B}'(\lambda)}{[\mathcal{F}(\lambda)+\mathcal{B}(\lambda)]^2} \geq 0.
\end{aligned}$$

That is, a higher sending rate along a path results in a higher loss rate. Since the sending rate along a path in the MP case is less than or equal to the sending rate of the SP case, given that these paths are homogeneous, the resulting average packet loss rate of MP will be less than or equal to that of SP. ∎

Let us now consider the conditional burst length distribution, of both SP and MP cases, conditioned on there being a loss. Let λ_1 be the mean streaming rate (in units of packets per second) along path 1 and $\delta_1 = 1/\lambda_1$ is the time between two consecutively transmitted packets. Then, in the SP case (as also derived in [2] for a voice-over-IP type application), the probability of having a packet error burst of size $m \geq 1$ is:

$$\mathrm{P}_{sp}[\text{error burst} = m] = \begin{cases} \pi_0(1)p_{0,1}^{(1)}(\delta_1)p_{1,0}^{(1)}(\delta_1) & \text{for } m = 1, \\ \pi_0(1)p_{0,1}^{(1)}(\delta_1)\left[p_{1,1}^{(1)}(\delta_1)\right]^{m-1} p_{1,0}^{(1)}(\delta_1) & \text{for } m \geq 2. \end{cases} \tag{6}$$

The probability of having a packet error burst of any size is therefore

$$\mathrm{P}_{sp}[\text{error burst}] = \sum_{m=1}^{\infty} \mathrm{P}_{sp}[\text{error burst} = m] = \pi_0(1)p_{0,1}^{(1)}(\delta_1).$$

Moreover, the conditional probability of having a packet error burst of size $m \geq 1$, conditioned on there being a loss, is equal to

$$\begin{aligned} \mathrm{P}_{sp}[\text{error burst of size } m|\text{ error burst}] &= \frac{\mathrm{P}_{sp}[\text{error burst} = m]}{\mathrm{P}_{sp}[\text{error burst}]} \\ &= \left[p_{1,1}^{(1)}(\delta_1)\right]^{m-1} p_{1,0}^{(1)}(\delta_1) \qquad \text{for } m \geq 1. \end{aligned} \tag{7}$$

In the MP case, let us consider the special case of DPRR streaming, i.e., $N = 2$. Let λ_2 be the streaming rate (in units of packets per second) along path 1 or path 2. Note that under DPRR, $\lambda_2 = \lambda_1/2$. Then, the time between two consecutively transmitted packets along the same path is $\delta_2 = 1/\lambda_2 = 2\delta_1$. To understand the basic tradeoff between SP and MP streaming, we also assume that both paths are homogeneous such that they are characterized by a stationary continuous time Gilbert model of the same parameters (i.e., $\mu_0(1) = \mu_0(2)$ and $\mu_1(1) = \mu_1(2)$). Given this simplification, the stationary distributions for both paths are the same (i.e., $\pi_0(1) = \pi_0(2)$; $\pi_1(1) = \pi_1(2)$) and we can express all performance measures using the parameters of path 1. Under these assumptions, the probability of having a packet error burst of size $m \geq 1$ is:

$$\mathrm{P}_{dp}[\text{error burst} = m] = \begin{cases} \pi_0(1)\pi_1(1)p_{0,0}^{(1)}(2\delta_1) & \text{for } m = 1, \\ \pi_0(1)\pi_1(1)\left[p_{1,1}^{(1)}(2\delta_1)\right]^{m-2} p_{0,1}^{(1)}(2\delta_1)p_{1,0}^{(1)}(2\delta_1) & \text{for } m \geq 2. \end{cases} \tag{8}$$

and the probability of having a packet error burst of any size is therefore:

$$\begin{aligned} \mathrm{P}_{dp}[\text{error burst}] &= \sum_{m=1}^{\infty} \mathrm{P}_{dp}[\text{error burst} = m] \\ &= \pi_0(1)\pi_1(1)p_{0,0}^{(1)}(2\delta_1) + \sum_{m=2}^{\infty} \pi_0(1)\pi_1(1)[p_{1,1}^{(1)}(2\delta_1)]^{m-2}p_{0,1}^{(1)}(2\delta_1)p_{1,0}^{(1)}(2\delta_1) \\ &= \pi_0(1)\pi_1(1)\left[p_{0,0}^{(1)}(2\delta_1) + \frac{p_{0,1}^{(1)}(2\delta_1)p_{1,0}^{(1)}(2\delta_1)}{1 - p_{1,1}^{(1)}(2\delta_1)}\right] = \pi_0(1)\pi_1(1)\left[p_{0,0}^{(1)}(2\delta_1) + p_{0,1}^{(1)}(2\delta_1)\right] \\ &= \pi_0(1)\pi_1(1). \end{aligned}$$

Then, the conditional probability of having a packet error burst of size $m \geq 1$, conditioned on there being a packet error, is equal to:

$$\begin{aligned} \mathrm{P}_{dp}[\text{error burst of size } m|\text{ error burst}] &= \frac{\mathrm{P}_{dp}[\text{error burst} = m]}{\mathrm{P}_{dp}[\text{error burst}]} \\ &= \begin{cases} p_{0,0}^{(1)}(2\delta_1) & \text{for } m = 1, \\ \left[p_{1,1}^{(1)}(2\delta_1)\right]^{m-2} p_{0,1}^{(1)}(2\delta_1)p_{1,0}^{(1)}(2\delta_1) & \text{for } m \geq 2. \end{cases} \end{aligned} \tag{9}$$

We can now state the conditions under which the DPRR approach will have a small conditional burst error than the SP approach. Before we present this result, let us present the definition and a basic lemma of stochastic comparison [17].

Definition 1 *We say that the random variable X is stochastically larger than the random variable Y, written $X \geq_{st} Y$, if $P[X \geq z] \geq P[Y \geq z]$ for all z.*

Lemma 1 *We say that $X \geq_{st} Y$ iff $E[f(X)] \geq E[f(Y)]$ for all increasing functions f.*

Now, let $\mathcal{B}_{sp}$ and $\mathcal{B}_{dp}$ be the random variables representing the conditional packet error burst size, given that there is at least one packet error, under the SP and the homogeneous DPRR approaches, respectively. Then, we have the following result.

Theorem 2 *If $p_{0,1}(2\delta_1)p_{1,0}(2\delta_1) \leq p_{1,1}(\delta_1)p_{1,0}(\delta_1)$, then $\mathcal{B}_{sp} \geq_{st} \mathcal{B}_{dp}$.*

Proof: First, note that $p_{1,1}(t)$ is an non-increasing function of t. If $p_{0,1}(2\delta_1)p_{1,0}(2\delta_1) \leq p_{1,1}(\delta_1)p_{1,0}(\delta_1)$, then from Equations (7) and (9), we can deduce that

$$\mathrm{P}_{dp}[\text{error burst of size } m | \text{ error burst}] \leq \mathrm{P}_{sp}[\text{error burst of size } m | \text{ error burst}] \qquad m \geq 2.$$

Since

$$\begin{aligned}
\sum_{m=1}^{\infty} P_{sp}[\mathcal{B}_{sp} = m] &= \sum_{m=1}^{\infty} P_{dp}[\mathcal{B}_{dp} = m] = 1 \qquad \text{and} \\
\sum_{m=j}^{\infty} P_{sp}[\mathcal{B}_{sp} = m] &\geq \sum_{m=j}^{\infty} P_{dp}[\mathcal{B}_{dp} = m] \qquad \text{for } j \geq 2,
\end{aligned}$$

we can conclude that $\mathcal{B}_{sp} \geq_{st} \mathcal{B}_{dp}$. ∎

Remark: Note that $\mathcal{B}_{sp} \geq_{st} \mathcal{B}_{dp}$ implies (based on Lemma 1) that $E[f(\mathcal{B}_{sp})] \geq E[f(\mathcal{B}_{dp})]$ for all increasing functions f. Therefore, we can conclude that for all moments of $\mathcal{B}_{sp}$ and $\mathcal{B}_{dp}$, we have $E[\mathcal{B}_{sp}^k] \geq E[\mathcal{B}_{dp}^k]$ for $k \geq 1$, where $E[\mathcal{B}_{sp}^k]$ and $E[\mathcal{B}_{dp}^k]$ refer to the k^{th} moments of $\mathcal{B}_{sp}$ and $\mathcal{B}_{dp}$, respectively. The implication of the above theorem is that the homogeneous DPRR approach will have a lower mean conditional burst length than the SP approach, given that the theorem's condition is satisfied.

Let us now consider the lag-1 autocorrelation of packet errors metric. We begin with the SP approach. The lag-1 autocorrelation function $R[X_t X_{t+\delta_1}]$ measures the degree of dependency of consecutive packet errors. For example, a high positive value of $R[X_t X_{t+\delta_1}]$ implies that a lost packet is very likely to be followed by another lost packet. On the other hand, a high negative value of $R[X_t X_{t+\delta_1}]$ implies that a lost packet is likely to be followed by a successful packet arrival. Also, if the statistics of the consecutive packet losses are not correlated[2], then $R[X_t X_{t+\delta_1}] = 0$.

The lag-1 autocorrelation for the SP approach is

$$R[X_t X_{t+\delta_1}] = \frac{E[(X_t - \overline{X})(X_{t+\delta_1} - \overline{X})]}{E[(X_t - \overline{X})^2]} = \frac{E[X_t X_{t+\delta_1} - \overline{X}^2]}{E[X_t^2 - \overline{X}^2]}.$$

Since $\overline{X} = \pi_1(1) = \mu_0(1)/[\mu_0(1) + \mu_1(2)]$, $E[X_t X_{t+\delta_1}] = \pi_1(1) p_{1,1}^{(1)}(\delta_1)$ and $E[X_t^2] = \pi_1(1) = \mu_0(1)/[\mu_0(1) + \mu_1(2)]$, substituting these expressions into the above equation, gives us

$$R[X_t X_{t+\delta_1}] = \frac{\pi_1(1) p_{1,1}^{(1)}(\delta_1) - \pi_1^2(1)}{\pi_1(1)[1 - \pi_1(1)]} = \frac{[\mu_0(1) + \mu_1(1)] p_{1,1}^{(1)}(\delta_1) - \mu_0(1)}{\mu_1(1)}. \tag{10}$$

Lemma 2 *For a high (low) bandwidth streaming application, the lag-1 autocorrelation of the SP streaming approach is positively correlated (tends to zero).*

[2] Note that if the lag-1 autocorrelation, $R[X_t X_{t+\delta_1}]$, is equal to 0, it does not necessarily imply that consecutive packet losses are not correlated.

Proof: Note that when $\delta_1 \to 0$, $p_{1,1}^{(1)}(\delta_1) \to 1$, and consequently the lag-1 autocorrelation $R[X_t X_{t+\delta_1}]$ approaches 1. In other words, if the streaming application has a high bandwidth requirement such that the inter-packet spacing tends to zero, then the consecutive packet losses are *"positively"* correlated. On the other hand, when $\delta_1 \to \infty$, $p_{1,1}^{(1)}(\delta_1) \to \mu_0(1)/[\mu_0(1)+\mu_1(1)]$, and consequently the lag-1 autocorrelation $R[X_t X_{t+\delta_1}] \to 0$. This implies that for low bandwidth streaming applications, wherein the inter-packet spacing is very large, the lag1-autocorrelation tends to zero. ∎

Let us also derive the lag-1 autocorrelation of the homogeneous DPRR approach. The lag-1 autocorrelation in this case is:

$$E[X_t^{(1)} X_{t+\delta_1}^{(2)}] = \frac{E[(X_t^{(1)} - \overline{X^{(1)}})(X_{t+\delta_1}^{(2)} - \overline{X^{(2)}})]}{\sqrt{E[(X_t^{(1)} - \overline{X^{(1)}})^2]E[(X_t^{(2)} - \overline{X^{(2)}})^2]}}. \tag{11}$$

Because both paths are homogeneous (i.e., their respective Gilbert models have the same parameters), we can simplify the above expression as:

$$\begin{aligned} E[X_t^{(1)} X_{t+\delta_1}^{(2)}] &= \frac{E[X_t^{(1)} X_{t+\delta_1}^{(2)} - \overline{X^{(1)}}^2]}{E[(X_t^{(1)} - \overline{X^{(1)}})^2]} = \frac{E[X_t^{(1)} X_{t+\delta_1}^{(2)}] - E[\overline{X^{(1)}}^2]}{E[\left(X_t^{(1)}\right)^2] - E[\left(\overline{X^{(1)}}\right)^2]} \\ &= \frac{\left(\frac{\mu_1(1)}{\mu_0(1)+\mu_1(1)}\right)\left(\frac{\mu_1(2)}{\mu_0(2)+\mu_1(2)}\right) - \left(\frac{\mu_1(1)}{\mu_0(1)+\mu_1(1)}\right)^2}{\mu_0(1)\mu_1(1)/\left(\mu_0(1)+\mu_1(1)\right)^2} = \frac{\left(\frac{\mu_1(1)}{\mu_0(1)+\mu_1(1)}\right)^2 - \left(\frac{\mu_1(1)}{\mu_0(1)+\mu_1(1)}\right)^2}{\mu_0(1)\mu_1(1)/\left(\mu_0(1)+\mu_1(1)\right)^2} \\ &= 0 \end{aligned} \tag{12}$$

In fact, we can see that the consecutive packet losses under the homogeneous DPRR application are *"uncorrelated"* since we have assumed independence of the two paths.

2.4 Performance Analysis of SP vs. Multi-path Streaming (with FEC)

We have shown that loss characteristics can be improved with multi-path streaming as compared to single path streaming, under conditions and metrics specified above. However, an interesting question that remains is whether there are still benefits to be gained once some form of redundancy is added to the stream. Specifically, we consider the use of an erasure code (as defined below), to which we will refer as FEC in the remainder of the paper. Hence, in this section we focus on the basic understanding of the performance of single path vs. multi-path streaming when FEC is added to the stream.

Since numerous coding schemes exist, we first give the details of the simple FEC scheme considered here. We divide a video file into groups of data packets such that each group consists of k data packets. Given each group of k data packets, we generate $n > k$ packets. We refer to these n packets as a FEC group. The encoding scheme is such that, if the number of lost packets within a FEC group is less than or equal to $(n-k)$, then we can reconstruct the original k data packets within that FEC group.

Let us first derive the average packet loss rate under the SP approach. As before, assume that we use path 1 which is characterized by a Gilbert model, as defined above, with parameters $\mu_0(1)$ and $\mu_1(1)$. The streaming application generates packets at a rate of λ (in unit of packet/sec)[3].

[3]Note that here, "packets" includes both data packets and packets carrying redundant information.

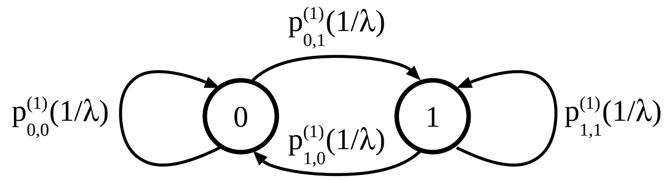


Figure 2: An Embedded Markov Chain which describes whether a transmitted packet is loss or not.

Whenever a packet is transmitted along this path, it may be lost (if the state of the path is "1") or it may arrive successfully at the receiver (if the state of the path is "0"). Figure 2 depicts an embedded Markov chain of this path wherein the two consecutive embedded points are $1/\lambda$ units apart. The derivation of transition probabilities of this DTMC is based on Equation (1); hence they are a function of the Gilbert model's parameters $\mu_0(1)$ and $\mu_1(1)$ as well as the packet transmission rate λ. The steady state probabilities of this embedded Markov chain are $\pi_0(1) = \frac{\mu_1(1)}{\mu_0(1)+\mu_1(1)}$ and $\pi_1(1) = \frac{\mu_0(1)}{\mu_0(1)+\mu_1(1)}$.

We are now interested in deriving $P^{(1)}(j,n)$, which is the probability of losing j packet in an n packet transmission. We define

$$P_i^{(1)}(j,n) = \text{Prob}(j,n|\text{initial state of the path is } i) \qquad i \in \{0,1\}$$

as the probability of j lost packet in an n packet transmission, given that the *first* packet was transmitted when the path was in state i (where $i \in \{0,1\}$). We then have:

$$P^{(1)}(j,n) = P_0^{(1)}(j,n)\pi_0(1) + P_1^{(1)}(j,n)\pi_1(1) \qquad j = 0,1,\ldots,n. \tag{13}$$

We also define:

$$\begin{aligned} L_i^{(1)}(j,n) &= \text{Prob}(j,n|\text{the initial state of the path is } i \text{ and the final state is } 0) \qquad i \in \{0,1\} \\ H_i^{(1)}(j,n) &= \text{Prob}(j,n|\text{the initial state of the path is } i \text{ and the final state is } 1) \qquad i \in \{0,1\} \end{aligned}$$

where $L_i^{(1)}(j,n)$ ($H_i^{(1)}(j,n)$) is the probability that we have j lost packets in an n packet transmission, given that the *first* packet was transmitted when the path was in state i (where $i \in \{0,1\}$) and that the *last* packet was transmitted when the path was in state 0 (state 1). Then we have:

$$P_i^{(1)}(j,n) = L_i^{(1)}(j,n) + H_i^{(1)}(j,n) \qquad i \in \{0,1\} \text{ and } j = 0,1,\ldots,n. \tag{14}$$

We can also express $L_i^{(1)}(j,n)$ and $H_i^{(1)}(j,n)$ in the following recursive forms:

$$L_i^{(1)}(j,n) = L_i^{(1)}(j,n-1)(1-p_{0,1}^{(1)}(1/\lambda)) + H_i^{(1)}(j,n-1)p_{1,0}^{(1)}(1/\lambda) \qquad j < n, \tag{15}$$

$$H_i^{(1)}(j,n) = L_i^{(1)}(j-1,n-1)p_{0,1}^{(1)}(1/\lambda) + H_i^{(1)}(j-1,n-1)(1-p_{1,0}^{(1)}(1/\lambda)) \qquad j < n. \tag{16}$$

where we also have the following boundary conditions:

$$L_i^{(1)}(j,m) = 0 \qquad i \in \{0,1\};\ j = 0,1,\ldots,n \text{ and } m \le j \tag{17}$$

$$L_0^{(1)}(0,m) = (1-p_{0,1}^{(1)}(1/\lambda))^{m-1} \qquad \text{for } m = 1,2,\ldots,n \tag{18}$$

$$L_1^{(1)}(0,m) = 0 \qquad \text{for } m = 1,2,\ldots,n \tag{19}$$

$$
\begin{aligned}
H_i^{(1)}(j,m) &= 0 && i \in \{0,1\}; j = 1,2,\ldots,n \text{ and } m < j && (20)\\
H_i^{(1)}(0,m) &= 0 && \text{for } i \in \{0,1\} \text{ and } m = 0,1,\ldots,n && (21)\\
H_0^{(1)}(m,m) &= 0 && \text{for } m = 1,2,\ldots,n && (22)\\
H_1^{(1)}(m,m) &= (1 - p_{1,0}^{(1)}(1/\lambda))^{m-1} && \text{for } m = 1,2,\ldots,n. && (23)
\end{aligned}
$$

Remark: To compute the value of $P^{(1)}(j,n)$ in Equation (13), we need to compute the values of the four square matrices $\boldsymbol{L}_0^{(1)}, \boldsymbol{L}_1^{(1)}, \boldsymbol{H}_0^{(1)}$, and $\boldsymbol{H}_1^{(1)}$, whose entries can be computed using Equations (15) through (23). Each of these matrices is of size $(n+1) \times (n+1)$. In other words, computing the values of $P^{(1)}(j,n)$ (for all j) has a computational complexity of $\Theta(4(n+1)^2)$.

Let P_{sp} be the probability of an irrecoverable error within a FEC group. It is equal to

$$
\begin{aligned}
P_{sp} = \sum_{j=n-k+1}^{n} P^{(1)}(j,n) &= \sum_{j=n-k+1}^{n} \left[P_0^{(1)}(j,n)\pi_0(1) + P_1^{(1)}(j,n)\pi_1(1)\right]\\
= \sum_{j=n-k+1}^{n} &\left[\left(L_0^{(1)}(j,n) + H_0^{(1)}(j,n)\right)\left(\frac{\mu_1(1)}{\mu_0(1)+\mu_1(1)}\right) + \right.\\
&\left. + \left(L_1^{(1)}(j,n) + H_1^{(1)}(j,n)\right)\left(\frac{\mu_0(1)}{\mu_0(1)+\mu_1(1)}\right)\right].
\end{aligned}
$$

To derive the average data packet loss rate (with use of FEC) for the SP approach, denoted by $\mathcal{L}_{sp}$, we consider the following two cases, based on the number of lost packets, $j \in \{0,1,\ldots,n\}$, within a FEC group.

Case 1: $j \leq n-k$
If j, the number of lost packet within a FEC group, is less than or equal to $n-k$, then all k data packets can be reconstructed at the receiver. Hence, this case does not contribute to *information* loss and $\mathcal{L}_{sp} = 0$.

Case 2: $j > n-k$
In this case, the lost data packets cannot be fully reconstructed and some information will be lost. However, given that there j lost packets within a FEC group, there are a number of different ways to distribute these losses among the n packets of the FEC group. To understand this effect, let us illustrate it using an example. Assume that $n = 5$ and $k = 4$. If $j = 2$, then there are two possible ways to distribute these two lost packets among the packets of the FEC group: (1) the two lost packets are the data packets within the FEC group, or (2) one lost packet is a data packet and the other lost packet corresponds to redundant information in the FEC code. In the first case, we lost 2 data packets out of a 4 data packet transmission. In the second case, we lost 1 data packet out of a 4 data packet transmission. Using the same argument, if $j = 5$, then there is only one way to distribute these five lost packets among packets of the FEC group. That is, all data packets are lost. Therefore, given that there are j lost packets, the number of ways to distribute the j lost packets among the packets of a FEC group is $\mathcal{W} = \mathcal{M} - j + (n-k) + 1$ where $\mathcal{M} = \min\{j,k\}$. Let $\mathcal{L}(j)$ be the average data packet loss rate given that there are j lost packets in a FEC group. Then, we have

$$
\begin{aligned}
\mathcal{L}(j) &= \frac{1}{\mathcal{W}} \sum_{i=j-(n-k)}^{\mathcal{M}} \frac{i}{k}\\
&= \left(\frac{1}{\mathcal{M} - j + (n-k) + 1}\right)\left(\frac{1}{k}\right)\left(\frac{\mathcal{M}(\mathcal{M}+1)}{2} - \frac{(j-(n-k))(j-(n-k)-1)}{2}\right) \quad (24)
\end{aligned}
$$

It is now easy to derive $\mathcal{L}_{sp}$, the average data packet loss rate (with the use of FEC) for the SP approach as follows:

$$\begin{aligned}
\mathcal{L}_{sp} &= \sum_{j=n-k+1}^{n} P^{(1)}(j,n)\mathcal{L}(j) \\
&= \sum_{j=n-k+1}^{n} \left[P_0^{(1)}(j,n)\pi_0(1) + P_1^{(1)}(j,n)\pi_1(1)\right]\mathcal{L}(j) \\
&= \sum_{j=n-k+1}^{n} \Bigg[\bigg(L_0(j,n) + H_0(j,n)\bigg)\left(\frac{\mu_1(1)}{\mu_0(1)+\mu_1(1)}\right)\mathcal{L}(j) \; + \\
&\qquad \bigg(L_1^{(1)}(j,n) + H_1^{(1)}(j,n)\bigg)\left(\frac{\mu_0(1)}{\mu_0(1)+\mu_1(1)}\right)\mathcal{L}(j)\Bigg]. \qquad (25)
\end{aligned}$$

To derive the average data packet loss rate (with use of FEC) for the MP approach, let us first consider a simple case of dual-path streaming. Assume that there are two servers S_1 and S_2 that use two different, possibly heterogeneous, paths. We use the same FEC scheme as described above to generate a stream of data divided into n packet FEC groups. To transmit the packets within a FEC group, server S_1 transmits n_1 packets while server S_2 transmits n_2 packets such that $n_1 + n_2 = n$. Based on the similar argument we made above in the SP case, we have

$$P^{(1)}(j,n_1) = P_0^{(1)}(j,n_1)\pi_0(1) + P_1^{(1)}(j,n_1)\pi_1(1) \qquad j = 0,1,\ldots,n_1 \qquad (26)$$
$$P^{(2)}(j,n_2) = P_0^{(2)}(j,n_2)\pi_0(2) + P_1^{(2)}(j,n_2)\pi_1(2) \qquad j = 0,1,\ldots,n_2. \qquad (27)$$

The computation of $P_i^{(h)}(j,n_h)$ where $i \in \{0,1\}$ and $h \in \{1,2\}$ is similar to the approach mentioned above, that is, by evaluating the entries of the corresponding four matrices. The computational complexity would then be $\Theta(4(n_1+1)^2 + 4(n_2+1)^2)$.

Let P_{2p} be the probability of an irrecoverable error within a FEC group. It is equal to

$$P_{2p} = \sum_{j=n-k+1}^{n} \sum_{h=0}^{j} P^{(1)}(h,n_1)P^{(2)}(j-h,n_2), \qquad (28)$$

which involves a convolution between the two probability mass functions, $P^{(1)}(j,n_1)$ and $P^{(2)}(j,n_2)$. Let $\mathcal{L}_{2p}$ be the average data packet loss rate (with use of FEC) for the dual path approach. Then, we have

$$\mathcal{L}_{2p} = \sum_{j=n-k+1}^{n} \sum_{h=0}^{j} P^{(1)}(h,n_1)P^{(2)}(j-h,n_2)\mathcal{L}(j). \qquad (29)$$

In general, if we employ N servers $S_1, S_2, \ldots, S_N$, then the probability of an irrecoverable error within a FEC group is

$$P_{Np} = \sum_{j=n-k+1}^{n} \left(\sum_{i_1+\ldots+i_N=j} P^{(1)}(i_1,n_1)P^{(2)}(i_2,n_2)\cdots P^{(N)}(i_N,n_1) \right). \qquad (30)$$

The average data packet loss rate with FEC under a MP streaming with N paths is

$$\mathcal{L}_{Np} = \sum_{j=n-k+1}^{n} \left(\sum_{i_1+\ldots+i_N=j} P^{(1)}(i_1,n_1)P^{(2)}(i_2,n_2)\cdots P^{(N)}(i_N,n_1) \right)\mathcal{L}(j). \qquad (31)$$

In the case of the other two performance measures, namely, the conditional burst length distribution and the lag-1 autocorrelation, we resort to the use of simulation, as described in the following section.

3 Analytical Model Based Evaluation

In this section, we further evaluate the loss characteristics of the SP vs. MP methods using simulations of the Gilbert model described in Section 2. The simulations allow us to consider the loss characteristics under more sophisticated scenarios than in Section 2. Specifically, we assume an MPEG-1 video streaming application which generates packets at a rate of 120 packets per second with each packet containing 1400 bytes. We consider at most three senders (S_1, S_2, S_3) and one receiver C. Sender S_i uses path i to transmit its fraction of the data; unless otherwise stated, these paths are assumed to be independent. Moreover, in the figures given below (unless otherwise stated), the curves corresponding to SP streaming use path 1, the curves corresponding to MP streaming with 2 senders use paths 1 and 2, and the curves corresponding to MP streaming with 3 senders use all three paths. Unless stated otherwise, the packet assignment is carried out in a round-robin manner, e.g., if we use all three senders, then sender S_i transmits data packets at a rate of 40 packets per second. The loss process of path i is modeled by a continuous stationary Gilbert model (as defined in Section 2). Unless stated otherwise, we use $\mu_0(i) = 20$ and $\mu_1(i) = 70$, for $i = 1, 2, 3$. Lastly, we consider all the same performance metrics as defined in Section 2.

Experiment 1 (Data Loss Rate): In this experiment, we study the data packet loss rate of the SP and MP approaches, using only two paths, 1 and 2. The path parameters are as described above except that we vary the $\mu_0(2)$ parameter from 5 to 50. Table 2 illustrates the data loss rate for the single path(s) and the dual-path approaches (in each case, with and without the use of FEC, where the parameters for the FEC scheme are $n = 5$ and $k = 4$). We can observe that in this experiment:

- Without the use of FEC, the data packet loss rate of the dual path is approximately the mean of the data packet loss rates of paths 1 and 2. These results are consistent with the derivation of Section 2.
- With the use of FEC, (in this case $n = 5$ and $k = 4$), the achieved data packet loss rate can be *less* than the average of the data packet loss rates of the two corresponding single paths. This may occur due to the fact that error burst lengths in dual-path streaming tend to be shorter than in single-path streaming (refer Theorem 2 in Section 2), and hence a chance of recovery of lost data (using FEC) should also be higher.

This experiment also illustrates the potential advantages of multi-path streaming over "best path" streaming, even when losses (rather than throughput) are the important consideration. That is, when multiple paths are available (but throughput is not the issue), another approach might be to stream the data over the "best" available path (and as congestion conditions change keep switching the streaming of the data to the best available path at the time). Our experiment shows that MP streaming could provide better loss characteristics (e.g., when FEC is used) than the "best" available path. (Please refer to Experiment 6 below on further comparison to a best-path type approach.)

Experiment 2 (Data Loss Rate as a function of FEC parameters): In this experiment, we study the effects of FEC parameters on the data loss rate. In general, there are two ways to vary the FEC parameters. We can:

Loss rate: ($\mu_0(2)$)	single path: path 1 w/o FEC	single path: path 2 w/o FEC	dual-path without FEC	single path: path 1 with FEC	single path: path 2 with FEC	dual-path with FEC
5	0.221743	0.066767	0.144351	0.189053	0.053048	0.101264
15	0.221743	0.176153	0.199395	0.189053	0.147171	0.141632
20	0.221743	0.221743	0.222255	0.189053	0.189053	0.158861
35	0.221743	0.332848	0.278178	0.189053	0.297647	0.201947
50	0.221743	0.416609	0.319230	0.189053	0.385602	0.235681

Table 2: Data Loss rate with Heterogeneous Paths.

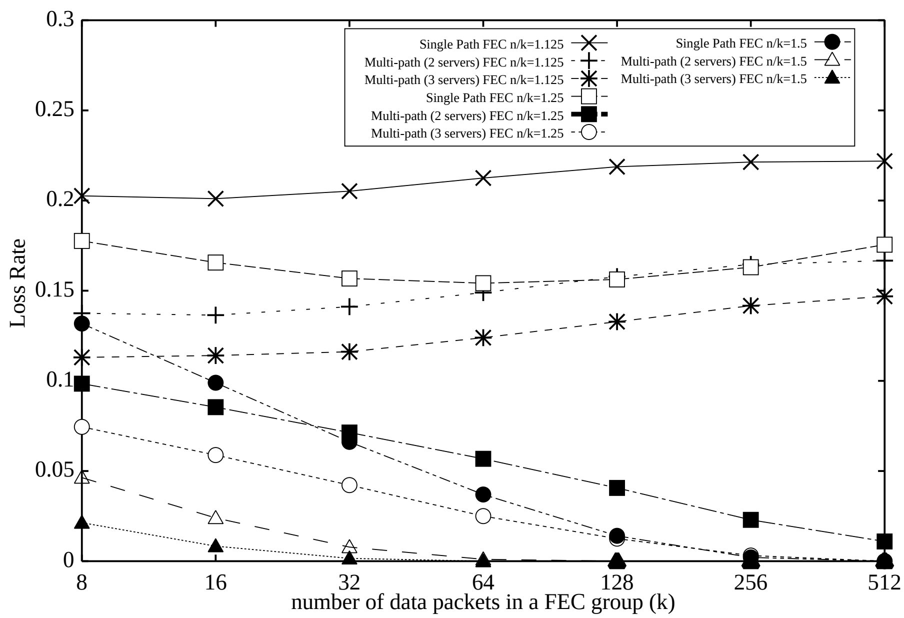


Figure 3: Loss rate as a function of n/k and k

1. Increase the degree of redundancy (e.g., for a given value of k, increase the value of n). Note that by increasing the degree of redundancy, we also increase the amount of traffic on the network.

2. Increase the values of n and k but keep the same ratio of n/k. This implies that we increase the FEC group size, and hence the application needs to maintain a larger receiving buffer (for reconstruction purposes in case of loss) as well as experience potentially higher latency (since a larger amount of information must be received prior to reconstruction of missing information).

Figure 3 illustrates the effects of FEC parameters on the data loss rate, and specifically, it depicts data loss rates for SP and MP streaming with $n/k = 1.125, 1.25$ and 1.5 as well as with different FEC group sizes (where we vary the number of data packets in a FEC group (k) from 8 to 512 packets). In this case the path parameters are $\mu_0(1) = 20$, $\mu_1(1) = 70$, $\mu_0(2) = \mu_0(3) = 10$, and $\mu_1(2) = \mu_1(3) = 80$. We observe that:

- Increasing the amount of redundancy (e.g., from $n/k = 1.125$ to 1.5) in SP or MP streaming can reduce the data loss rate. However, one can achieve a lower data packet loss rate with MP streaming with a smaller n/k ratio (as compared to SP streaming). In other words, without introducing additional network traffic, we can obtain better performance with MP streaming.

- Increasing the number of data packets in a FEC group (while keeping the same ratio of n/k) may not necessary reduce the data loss rate. For example, consider SP streaming; as we increase k, the data loss rate actually increases in some cases. The maybe explained by a possible "convergence" of the data loss rate, as a function of n and k, to a non-zero value (please refer to the Appendix for details).

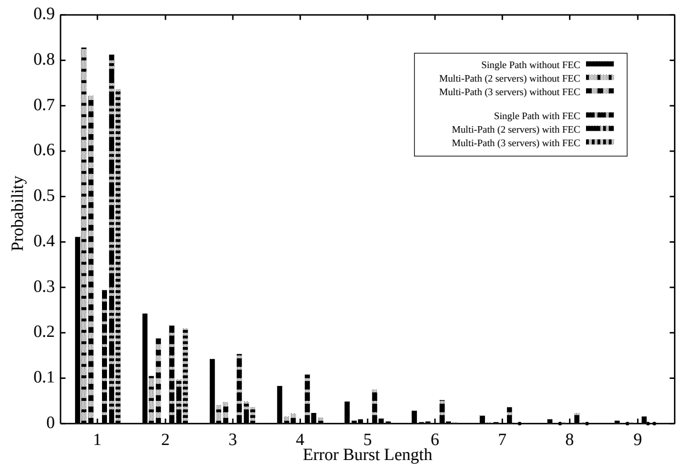


Figure 4: Conditional probability mass functions of error burst length.

Experiment 3 (Conditional Error Burst Length): In this experiment, we compare the conditional burst length distribution, conditioned on there being at least one error. Figure 4 illustrates the conditional probability mass functions of error burst length (as defined in Section 2). In this experiment, we observe that the packet error burst length is indeed stochastically less than the error burst length of the single path streaming. We also note, that the condition of Theorem 2 in Section 2 holds in this experiment[4]. This relationship also holds when we employ FEC.

Experiment 4 (Lag-1 Autocorrelation): In this experiment, we study the lag-1 autocorrelation of packet losses for both SP and MP streaming (as defined in Section 2). Figure 5 illustrates the lag-1 autocorrelation where $\mu_1(1) = \mu_1(2) = \mu_1(3) = 70$ and $\mu_0(i)$ is varied (identically) for all three paths. We make the following observations.

- When we use MP streaming without FEC, the lag-1 autocorrelation is nearly zero while the lag-1 autocorrelation of SP path streaming (with or without FEC) can be highly correlated.

- The use of FEC may increase the lag-1 autocorrelation (for both SP and MP approaches). This may be explained as follows. The irrecoverable losses (after the error correction process) are likely to end up "closer" in the resulting data stream than in the original data stream (one without the use of erasure codes), and hence the lag-1 autocorrelation in this new stream behaves similarly to lag-h autocorrelation of the original stream, where $h > 1$. However, we still observe that the lag-1 autocorrelation of MP streaming is significantly closer to zero as compared to SP streaming, even with the use of FEC.

[4]Note that here we illustrate the probability mass function rather than the probability distribution function, as we believe it depicts the results of the experiment better.

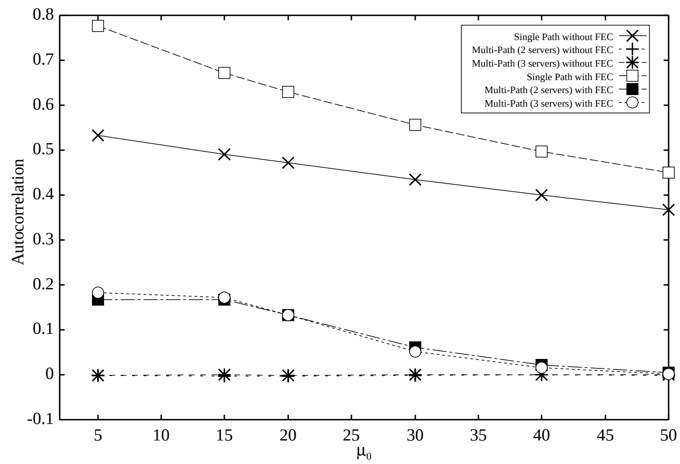
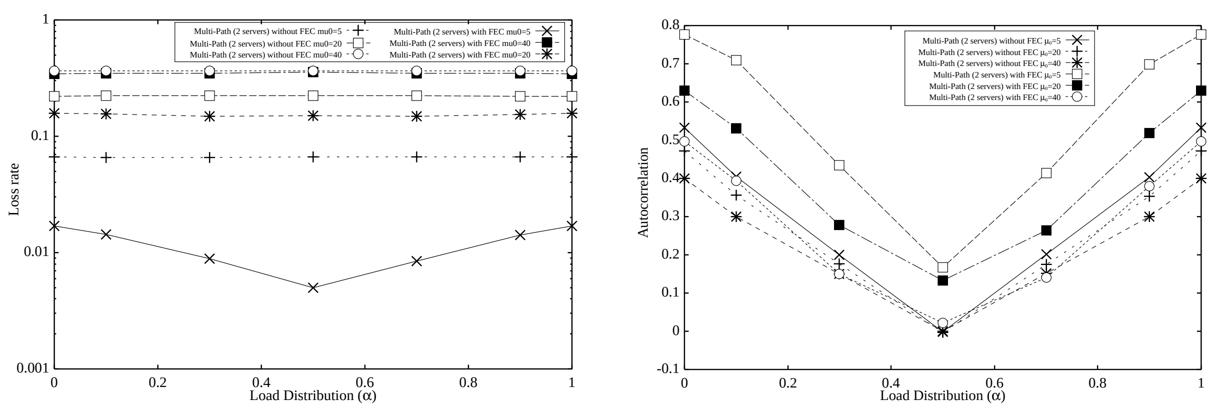


Figure 5: Lag-1 autocorrelation.

Experiment 5 (Effects of Load Distribution Among Senders): In previous experiments, all senders transmitted packets in a round-robin manner and hence the load distribution between all the senders was the same. In this experiment, we investigate effects of load distribution among senders. Specifically, we distribute the load among two senders only, where parameter α refers to



(a) Loss Rate

(b)Lag-1 Autocorrelation

Figure 6: Loss rate and Lag-1 autocorrelation for different load distributions for the dual-path streaming

the fraction of packets sent by sender 1. For instance, when $\alpha = 0.3$, sender 1 sends 30% of the packets while sender 2 sends 70% of packets. In the cases of $\alpha = 0$ and $\alpha = 1$, this degenerates to single path streaming using path 1 and path 2, respectively. Both path 1 and path 2 have the same parameters with $\mu_0 = 5, 20$, or 40 and μ_1 fixed at 70. Figure 6 illustrates results of this experiment. We observe that there is a slight improvement in loss rate when FEC is used and the load is equally distributed between the two senders. Moreover, in this experiment, the lag-1 autocorrelation reaches its minimum value under equal load distribution. This implies that simple round-robin packet distribution among paths should result in a higher quality of received video. That is, this simple approach of equal distribution is fairly robust.

Experiment 6 (Sensitivity Analysis): In this experiment, we study the relative performance of MP streaming vs. SP streaming when the SP streaming is performed over the *best* of the available paths. For example, if the performance metric is loss rate, then the path with the lowest loss rate is used. We note that implementation of this form of *best* single path streaming would likely require a fairly accurate monitoring of the loss characteristics of a path; otherwise, the wrong path might be selected. That is, the sensitivity (or robustness) of the streaming decisions to the accuracy of the available information about the network is an important issue.

In this sensitivity experiment, we consider a two-path system, where the fixed parameters are $\mu_0(1) = 20$ and $\mu_1(1) = \mu_1(2) = 70$ and $\mu_0(2)$ is varied from 5 to 50. In this scenario, the best-path approach believes (based on collected measurements) that path 2 is the better path (e.g., it may mis-estimate the $\mu_0(2)$ parameter as being less than 20). We vary $\mu_0(2)$ from 5 to 50, in order to see the effect of mis-estimation; hence, the best path approach *over-estimates* this parameter when the real value of $\mu_0(2)$ is less than 20 and *under-estimates* this parameter when the real value of $\mu_0(2)$ is greater than 20. We also consider a very *simple* MP streaming approach, where the load is distributed equally among the two senders in a round-robin manner (i.e., odd-numbered packets are sent along path 1 while even-numbered packets are sent along path 2).

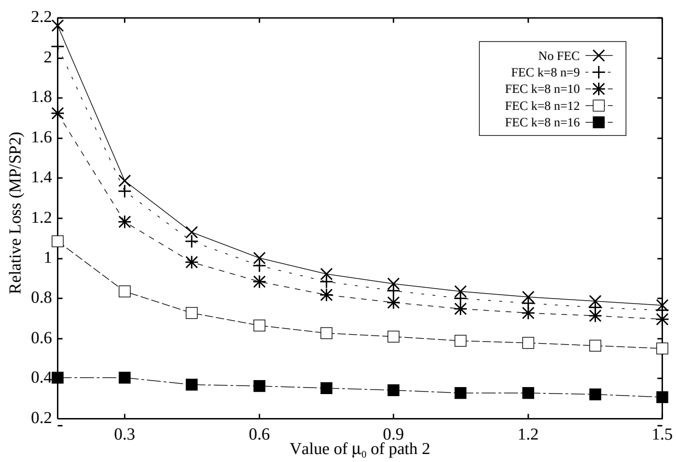
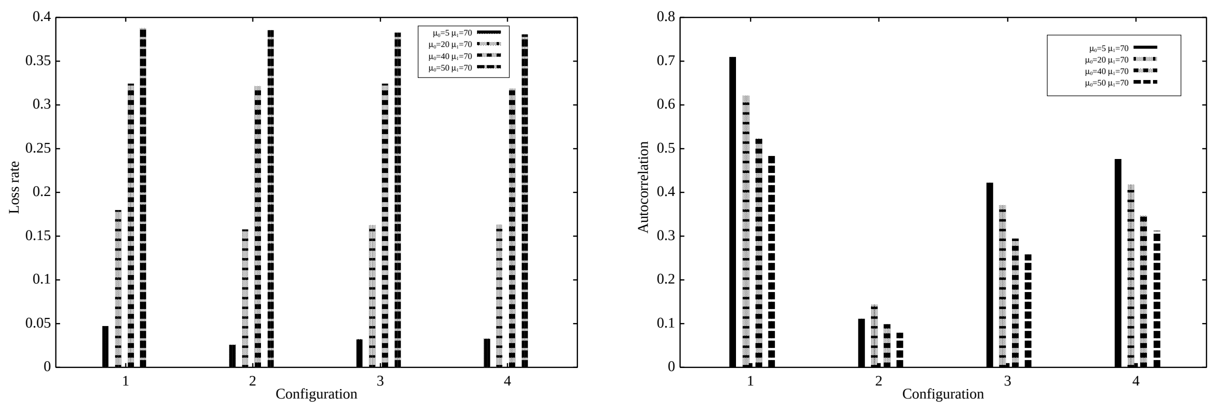


Figure 7: Relative loss of dual-path vs. single path when we vary $\mu_0(2)$ and FEC group size.

Figure 7 illustrates the *relative* loss rate (using several different FEC schemes) of the two approaches, which is defined as the data loss rate of dual-path streaming divided by the data loss rate of best-path streaming. Hence, a relative loss of less than 1 implies that the simple dual-path approach is doing better than the best path approach. In this figure, we observe that simple dual-path (round-robin) streaming does quite well compared to best-path streaming, even when there is significant differences in loss characteristics between the two paths. Of course, in cases where the best path has much better loss characteristics and with relatively little redundant information, the best-path approach has a lower data loss rate. However, we note that the best-path approach would require relatively accurate estimation of the path characteristics, which may be non-trivial especially as network conditions change. Hence, we believe that the MP approach is more robust as compared to best-path streaming.

Experiment 7 (Effects of Shared Points of Congestion on Various Performance Metrics): In this experiment, we study the effects of *shared points-of-congestion*, between the paths used by the different senders, on various performance measures. Senders S_1 and S_2 share the same point-of-congestion, which we can characterize by a Gilbert model (as defined in Section 2). Sender S_3 uses a path which does not share a point of congestion with S_1 and S_2 (as before, this path is characterized by a Gilbert model). All the application settings remain the same, and we consider the following four configurations.

- **Configuration 1:** Sender 1 is the only one streaming the data.
- **Configuration 2:** Senders 1 and 3 stream the data in a round-robin manner, i.e., each transmits at a rate of 60 data packets/second.
- **Configuration 3:** Senders 1, 2, and 3 stream the data in a round-robin manner, i.e., each transmits at a rate of 40 data packets/second.
- **Configuration 4:** Senders 1, 2, and 3 stream the data, but senders 1 and 2 transmit at a rate of 20 data packets/second while sender 3 transmits at a rate of 80 data packets/second.



(a) Data Loss Rate (b)Lag1-autocorrelation

Figure 8: Effects of shared points-of-congestion on data loss rate and lag-1 autocorrelation with FEC $(n = 10, k = 8)$.

Figure 8 illustrates the data loss rate and lag1-autocorrelation for above configurations, when FEC is used, with $(n = 10, k = 8)$. Moreover, we vary the $\mu_0(1), \mu_1(1), \mu_0(3)$, and $\mu_1(3)$ parameters (as described in the figure). From this figure, we observe the following.

- MP streaming (configuration 2, 3, and 4) has a lower data loss rate as compared to SP streaming (configuration 1).
- Detecting shared points of congestion is important, as including a greater number of paths in a transmission (under such conditions) may adversely affect the data loss rate. For example, equally splitting the workload among senders 1 and 3 (configuration 2) achieves a lower data loss rate than equally splitting the workload among senders 1, 2, and 3 (configuration 3). This occurs because senders 1 and 2 share the same point of congestion and with configuration 3 we are actually sending a greater fraction of the workload through this shared point of congestion. This agrees with intuition, as in this section we are effectively modeling a shared point of congestion as a single path/bottleneck, i.e., configuration 3 effectively corresponds to a configuration with two senders and an unequal split of workload between them.

- Of course, shared points of congestion adversely affect the lag-1 autocorrelation metric. For example, configuration 3 has a higher lag-1 autocorrelation than configuration 2. Again, the explanation given in the preceding point applies here as well.

4 Simulation Model Based Evaluation

In this section, we evaluate the performance of SP streaming vs. MP streaming using the NS-2 [8] simulator. NS-2 is a packet level simulator which allows us to study the performance measures (as defined in Section 2) under more realistic traffic and Internet protocols (such as UDP).

4.1 Simulation Setup

As in the previous section, we consider at most three senders (S_1, S_2 and S_3) and one receiver C. Figure 9 illustrates our simulation topology. Each sender transmits the video data, at a constant

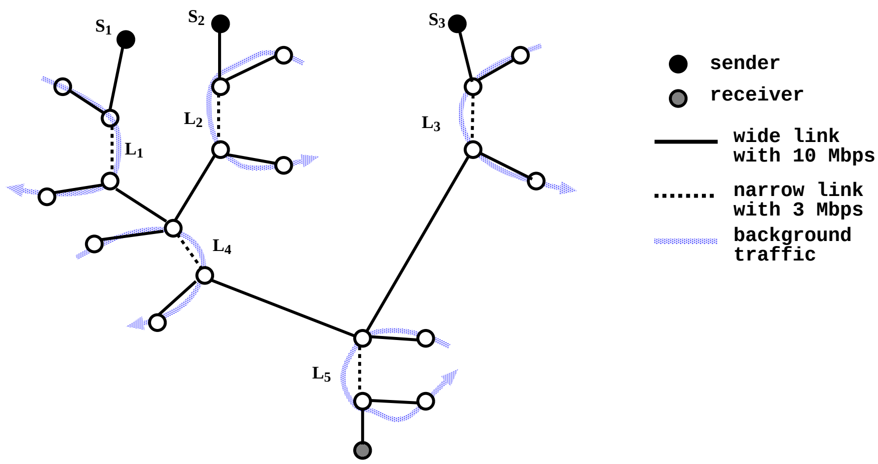


Figure 9: Simulation Topology.

rate, to the receiver C using the UDP protocol, with packet sizes of 1400 bytes. The data traffic goes through two types of links: (1) wide/higher capacity links (represented by solid lines) and (2) narrow/lower capacity links (represented by dotted lines). Each wide link has a bandwidth of 10 Mbps while the bandwidth of a narrow link is 3 Mbps. Each link has a different propagation delay and the propagation delay is generated using an exponential random variable with a mean of 200 ms. The streaming application has a sending rate of 1.5 Mbps which consumes 50% of the bandwidth of a narrow link. The actual sending rate of each sender is a function of the traffic load distribution. Unless stated otherwise, an equal distribution is used, e.g., for MP streaming with three senders (sending data in a round-robin manner), the sending rate of each sender is 0.5 Mbps. Background traffic (represented by grey arrows) is introduced at different narrow links. The background traffic is generated using exponential on/off sources. The average "on" time plus the average "off" time of these on/off sources is equal to 1 second. During the "on" times, the background source generates UDP traffic with a constant rate of 3 Mbps, which can saturate the capacity of the traversed narrow links. In the following experiments we vary the amount of "on" time within an average of 1 second period. For example, a background traffic rate of 1.8 Mbps represents an average "on" time of 0.6 seconds for an average of 1 second on/off period. There are three possible sets of background traffic locations. One set of local background traffic occurs on the narrow links L_i where $i = 1, 2, 3$. This background traffic competes with the corresponding sender S_i ($i = 1, 2, 3$) for the bandwidth resources of the narrow links L_1, L_2, and L_3, respectively. The second set of background traffic occurs on the narrow link L_4. This background traffic competes with senders S_1 *and* S_2 for the bandwidth resource of the narrow link L_4. The third set of background traffic occurs on the narrow link L_5. This background traffic competes with *all* three senders for the bandwidth resource of the narrow link L_5. Unless stated otherwise, SP streaming is done from sender 1 and dual-path streaming is done from senders 1 and 3.

Experiment 1 (Data Loss Rate): Figure 10 illustrates the data loss rates for SP and MP

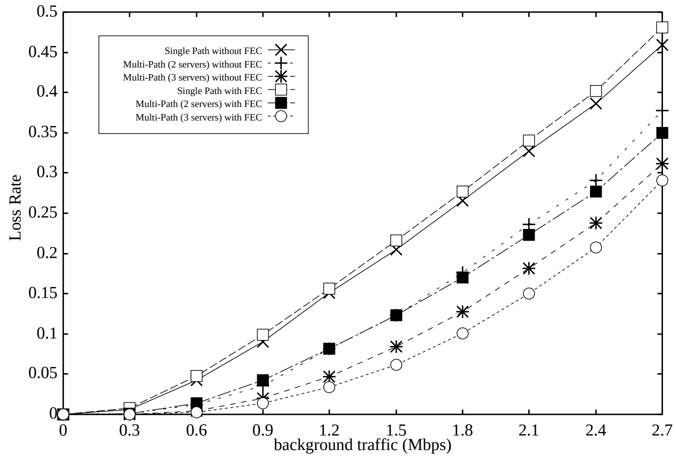


Figure 10: Loss rate with FEC parameters $n = 10$ and $k = 8$.

streaming. In this simulation, we vary the average background traffic through the narrow links L_1, L_2, and L_3 from 0 Mbps to 2.7 Mbps. (Note that the senders do not share points of congestion in this case.) From this figure, we observe the following. Firstly, MP streaming can achieve a significant reduction in the data loss rate as compared to SP streaming. Secondly, employment of FEC may actually increase the data packet loss rate; for example, the data loss rate of SP streaming with FEC is a bit higher than the data loss rate of SP without FEC. Thirdly, the improvements in the data loss rate achieved through the use of MP streaming *without* FEC is higher than that achieved through the use of FEC by adding it to SP streaming. This is potentially due to the fact that the use of FEC (with SP streaming) introduces additional traffic into the (already) congested network and hence results in higher data losses. On the other hand, the use of MP streaming achieves a significant reduction in data loss rate without introduction of additional network traffic.

Experiment 2 (Data Loss Rate as a function of FEC parameters): In this experiment, we study the effects of FEC parameters on the data loss rate. Again, we vary the FEC parameters as in Section 3. Figure 11 illustrates the data loss rate when a background traffic of 1.5 Mbps is used on each of the narrow links L_1, L_2, and L_3. We observe that:

- Increasing the degree of redundancy under SP streaming may not necessarily reduce the data loss rate, one reason being that introducing additional traffic (due to higher degree of redundancy) into an already congested network may result in higher packet loss rates. Hence, MP streaming may have a higher chance of decreasing the data loss rate with higher degrees of redundancy, i.e., with less traffic being introduced per path.

- MP streaming can significantly reduce data loss rate as compared to SP streaming.

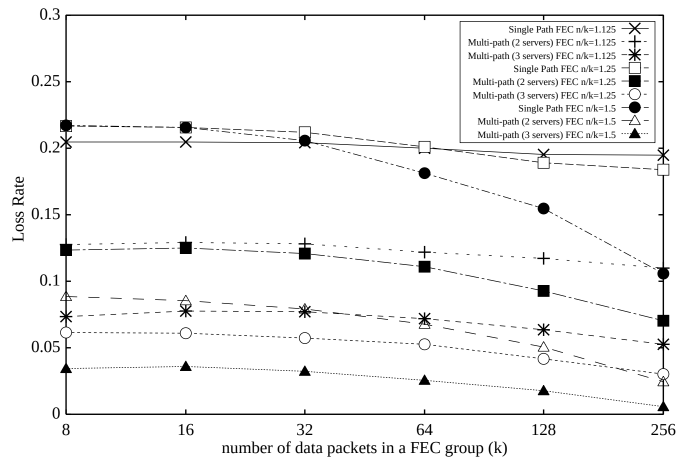


Figure 11: Loss rate as a function of n/k ratio and k

In summary, we observe that increasing the amount of redundancy (by increasing the n/k ratio) or increasing the FEC group size (and hence potentially suffering higher latency at the receiver with a need for larger buffer sizes) may not result in significant reduction in data loss rate, for either SP or MP streaming. On the other hand, taking advantage of multiple independent paths, can reduce the data loss rate significantly.

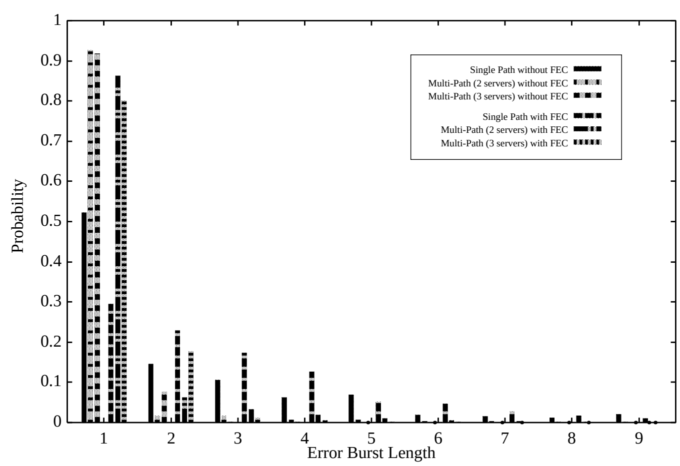


Figure 12: Conditional probability mass function for error burst length.

Experiment 3 (Conditional Error Burst Length): In this experiment, we compare the conditional burst length distribution, conditioned on there being at least one loss, of the SP and MP approaches. In this case a background traffic of 2.4 Mbps is used on each of the narrow links L_1, L_2 and L_3. The conditional probability mass function[5] of error burst length is given in Figure 12, where we observe that MP streaming has a stochastically smaller data packet burst length than SP streaming.

Experiment 4 (Lag-1 Autocorrelation): In this experiment, we study lag-1 autocorrelation of packet losses for both SP and MP streaming. Figure 13 illustrates the lag-1 autocorrelation as we

[5]As in Section 3 we illustrate the probability mass function rather than the probability distribution function, as we believe it depicts the results of the experiment better.

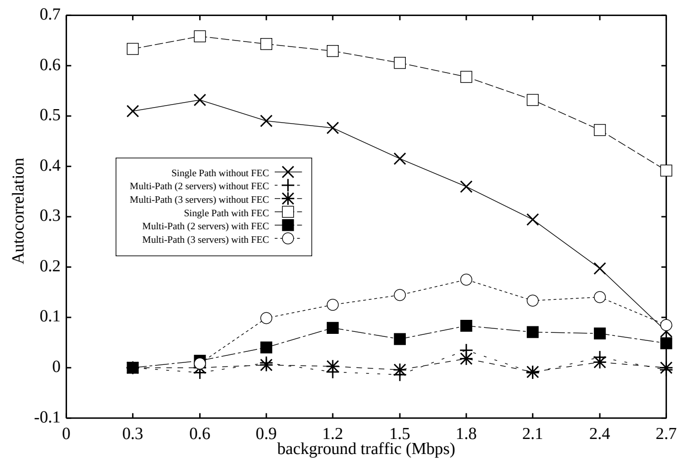


Figure 13: Lag-1 autocorrelation.

vary the background traffic on the narrow links L_1, L_2 and L_3. We observe the following.

- Without use of FEC, the MP lag-1 autocorrelation is close to zero (as derived in Section 2), i.e., the losses appear nearly uncorrelated when streaming over multiple independent paths. On the other hand, the correlation of losses with SP streaming can be quite high.
- With use of FEC, lag-1 autocorrelation may increase. We believe that a similar explanation (as given in Experiment 4 of Section 3) holds here. However, we still observe that the MP lag-1 autocorrelation is significantly lower than the SP lag-1 autocorrelation (under the same FEC scheme).

Lastly, the decrease in lag-1 autocorrelation as a function of higher background traffic may be counter-intuitive. One explanation may be that the "no losses" (i.e., the packets that are received successfully) in the resulting stream tend to be more "random" as congestion on the network increases.

Experiment 5 (Effects of Load Distribution among Senders): In previous experiments, all senders transmitted packets in a round-robin manner and hence the load on all senders (i.e., the amount of data streamed from each sender) was the same. In this experiment, we study the effects of different load distributions on the resulting loss characteristics observed at the receiver. Specifically, we consider the following configurations.

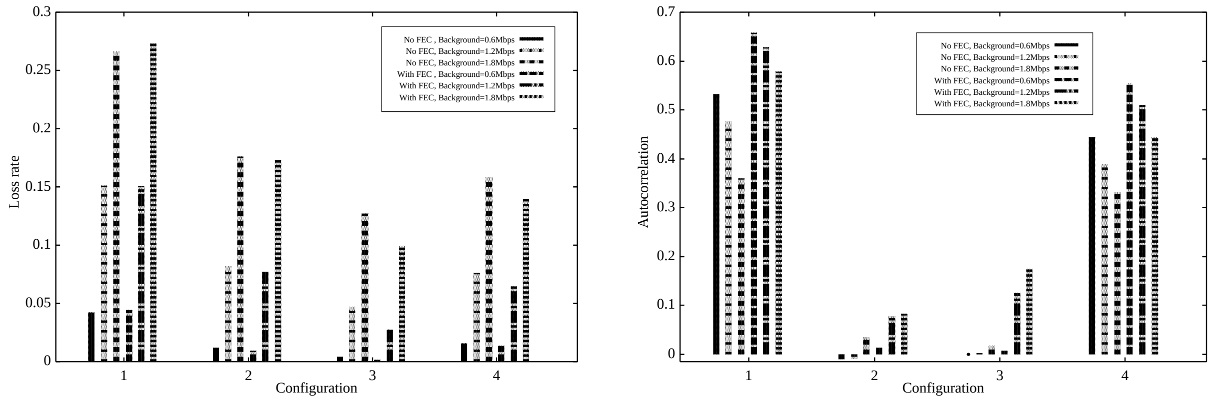
Configuration 1 Streaming from sender 1 only.

Configuration 2 Equal distribution of load between senders 1 and 3 only.

Configuration 3 Equal distribution of load among all senders.

Configuration 4 Sender 1 streams 1/6 of the data, sender 2 streams 1/6 of the data, and sender 3 streams 2/3 of the data.

Figure 14 depicts the data loss rate and the lag-1 autocorrelation of these configurations. In this experiment, equal distribution of load (configuration 3) tends to achieve a lower data loss rate and lag-1 autocorrelation.



(a) Loss Rate (b)Lag-1 Autocorrelation

Figure 14: Loss rate and Lag-1 autocorrelation under different load distributions

Experiment 6 (Sensitivity Analysis): In this experiment, we study the relative performance of MP streaming vs. SP streaming when the SP streaming is performed over the *best* of the available paths (please refer to Section 3 for a more detailed explanation of "best path" streaming and the motivation for making this comparison). Specifically, we consider a two senders system with only senders S_1 and S_3 transmitting packets. The background traffic on L_1 is fixed at 1.5 Mbps, and the background traffic on L_3 is varied from 0.3 to 2.7 Mbps. In this scenario, the best-path approach believes (based on collected measurements) that the path originating at sender S_3 experiences the least losses. Therefore, the best-path streaming approach always uses the path originating from sender S_3. We also consider a very *simple* MP streaming approach, which streams the data in a round-robin manner from S_1 and S_3.

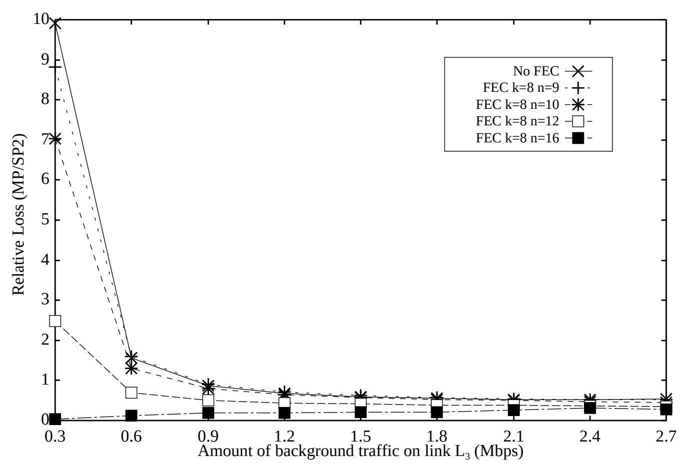
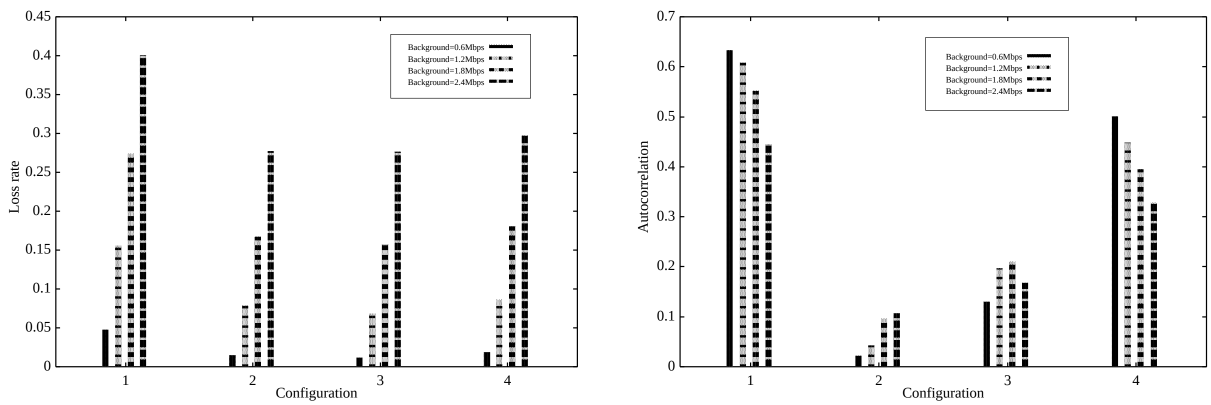


Figure 15: Relative Loss Rate when background traffic on link L_3 and FEC group size are varied.

Figure 15 illustrates the *relative* loss rate (using several different FEC schemes), which is defined as the data loss rate of dual-path streaming divided by the data loss rate of best-path streaming. Hence, a relative loss rate of less than 1, implies that simple dual-path streaming is more robust as compared to best-path streaming. As in Section 3, we observe that simple dual-path (round-robin) streaming does quite well compared to best-path streaming, even when there is significant differences in loss characteristics between the two paths. Of course, in cases where the best path has much better loss characteristics and with relatively little redundant information, the best-path approach has a lower data loss rate. Hence, we believe that the MP approach is more robust as compared to best-path streaming.

Experiment 7 (Effects of Shared Points of Congestion on Various Performance Metrics): In this experiment, we study the effects of *shared points-of-congestion*, between the paths used by the different senders, on various performance measures. Here, the background traffic is sent through the narrow links L_3 and L_4. Note that, having background traffic on L_4 implies that senders 1 and 2 share the same point-of-congestion. Again, we consider the four configurations described in Experiment 5 above.



(a) Data Loss Rate

(b)Lag1-autocorrelation

Figure 16: Effects of shared points-of-congestion on data loss rate and lag-1 autocorrelation with FEC $(n = 10, k = 8)$.

Figure 16 illustrates the data loss rate and lag-1 autocorrelation for these configurations, when FEC is used, with $(n = 10, k = 8)$. Moreover, we vary the background traffic on the two narrow links L_3 and L_4 among the following values: 0.6 Mbps, 1.2 Mbps, 1.8 Mbps, and 2.4 Mbps. From this figure, we observe the following.

- MP streaming (configurations 2, 3, 4) has a lower data loss rate as compared to SP streaming (configuration 1).
- Detecting shared points of congestion is important, as including a greater number of paths/senders (under such conditions) in the transmission may adversely affect the data loss rate.
- Shared points of congestion adversely affect the lag-1 autocorrelation metric. For example, configuration 3 has a higher lag-1 autocorrelation than configuration 2.

5 Other Considerations and Related Work

In this section we first briefly discuss some of the issues that should be explored when considering the use of MP streaming. We then survey related work on this topic.

5.1 Considerations in Use of MP Streaming

We note that one should also consider the potential costs or detrimental effects of multipath streaming. For instance, MP streaming might have an adverse effect on the resulting delay characteristics observed at the receiver. As a result, it might also require a large amount of receiver buffer space. In addition, the overheads associated with sending data over multiple paths and then assembling it into a single stream at the receiver should also be considered. Moreover, the overheads and complexity due to measurements needed to achieve better performance with MP streaming should also be considered. For instance, in our case, we employ detection of shared points of congestion [18] to improve the performance of our MP streaming system. Other approaches to MP streaming might require even more detailed information about the network (refer to Section 5.2) which is likely to result in a need for more "intrusive" and complex measurements. Lastly, scalability of such measurement schemes is an issue as well. However, the evaluation of such costs is outside the scope of this paper.

5.2 Related Work

We now give a brief survey of existing work on this topic, and specifically, we focus on those that either consider loss characteristics or can be deployed over best-effort networks (as these are considerations in our work as well). Earlier efforts on dealing with losses through the use of multiple independent paths (although at lower layers of the network) include dispersity routing, as proposed by Maxemchuk [11, 12, 13]. Briefly, a message is divided into a number of submessages which are then transmitted over a set of independent links in the network (and hence the number of submessages is limited by the number of such links). The focus in this work was on reducing delay, which includes reducing the number of retries needed to deliver a message without error, by sending the pieces of the data over multiple independent paths. Of course, addition of redundant information, where only a subset of the submessages would need to arrive correctly, is also possible under such a scheme. An important difference in our work is that we focus on streaming applications where the data transmission rate is determined by the application's needs rather than on delivering the data to its destination as fast as possible. Hence, in our case the data is sent through the network at a specific rate and that has an effect on loss characteristics, which we investigate here. Also, we do not consider retransmissions as there is usually little opportunity to retransmit data in such applications (due to their real-time constraints), and hence some amount of lossiness must be tolerated.

The use of multiple paths in routing data has of course been considered at the network layer. However, it is not generally done at the network layer in the current Internet. Hence, higher layer mechanisms should be considered. Another set of works on the topic considers higher level mechanisms, but requires some assistance from the lower layers and/or assumes significant knowledge of network topology and/or link capacities and delays (on all links used for data delivery). Given such knowledge, algorithms are proposed for selecting paths which can avoid congested routes. For instance, in [4], the authors focus on adaptation of delivery rate along the different paths, based on losses observed at the receiver. And, [3] considers proper scheduling of the initial portion of the video so as to reduce the start-up delay. In contrast, our approach does not rely on specific knowledge of topologies, capacities, delays, etc., and only considers whether a set of paths do or do not share joint points of congestion, as can be detected at the end-hosts. Moreover, our focus in this paper is on characterizing the benefits, with respect to loss characteristics, of a multipath approach as compared to a single path approach. Hence, our interest is in the more basic understanding of this problem.

Recent literature on this topic also includes works on voice-over-IP type applications. For instance, [10, 9] proposes a scheme for real-time audio transmission using multiple independent paths between a single sender and a single receiver, where multiple description coding (MDC) is used in multi-path delivery and a FEC approach is used in single-path delivery. These approaches are evaluated through simulation and experiments. In contrast, we believe that it is important to understand the effects of multi-path delivery on loss characteristics, even without the use of coding techniques. Hence, a great deal of our paper focuses on that. We also note that "live" applications (such as voice-over-IP) have different characteristics than pre-recorded applications (as we are considering here). For instance, one such difference is the need to disperse data in real-time, whereas in our case, we can distribute it to the multiple senders ahead of time; this makes application-level implementation simpler and possibly more efficient. Another difference might be the ability to address the potentially adverse effects of MP streaming on delay characteristics (as mentioned above).

The most recent work on the topic [16] is closest to ours in that it also considers delivery of pre-recorded video from multiple senders distributed across the network. However, this work focuses on a transport protocol as well as on optimization algorithms for (a) rate distribution among the paths (i.e., how much data to send over each path) and (b) packet distribution among the paths (i.e., which packet should be sent over which path), with the objective of minimizing the loss rate at the receiver. In an effort that will appear in the future [15][6] FEC techniques are added (as compared to [16]). Again, distribution algorithms are considered but with the objective of minimizing the probability of irrecoverable error. In contrast, due to the nature of the application, we believe that it is important to consider loss characteristics even when the losses cannot be fully recovered. That is, since we are considering delivery of video (which can be displayed even under some losses) in contrast to file transfer (which cannot tolerate losses), it is important to consider other metrics. As mentioned above, in this paper we consider, data loss rate (with and without the use of FEC), burst length distribution (with and without the use of FEC), as well as lag-1 autocorrelation (with and without the use of FEC), in our evaluation of potential benefits of multi-path streaming.

6 Conclusions

In this paper we investigated the potential benefits of an application-layer multi-path streaming approach to providing QoS over best-effort wide-area networks. As already mentioned, an advantage of this approach (as compared to approaches that require support of lower layers) is that the complexity of QoS provision can be pushed to the network edge and hence improve the scalability and deployment characteristics while at the same time provide a certain level of QoS guarantees. Our focus in this paper was on providing a fundamental understanding of the benefits of using multiple paths to deliver pre-recorded continuous media over best-effort wide-area networks, with loss characteristics being the main concern.

Our results indicate that in general, multi-path streaming exhibits better loss characteristics than single-path streaming (with or without use of an erasure code), which should result in a higher viewing quality of the received continuous media. These results can be used in guiding the design of multi-path continuous media systems. Overall, we believe that these results are quite

[6]This paper has not appeared yet, and hence we are referring to the version currently available on the authors' web page.

encouraging and warrant further study of multi-path streaming over wide-area networks. Our current and future efforts include: (a) investigation of potential benefits of streaming adaptation between multiple paths, as network conditions change as well as (b) validation of conclusions made here using real Internet experiments.

References

[1] R. E. Blahut. *Theory and Practice of Error Control Codes.* Addison Wesley, January 1983.

[2] Jean-Chrysostome Bolot, Sacha Fosse-Parisis, and Don Towsley. Adaptive FEC-Based Error Control for Internet Telephony. In *INFOCOM*, 1999.

[3] J.-C. Chen and S.-H. Chan. Multipath routing for video unicast over bandwidth-limited networks. In *the IEEE Globecom*, San Antonio, Texas, November 2001.

[4] H. Chu and K. Nahrstedt. Dynamic multi-path communication for video traffic. In *the Hawiian International Conference on System Science*, Hawaii, January, 1997.

[5] D. Le Gall. MPEG: a Video Compression Standard for Multimedia Applications. *Communications of the ACM*, April 1991.

[6] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Addison Wesley, 1993.

[7] Khaled Harfoush, Azer Bestravos, and John Byers. Robust identification of shared losses using end-to-end unicast probes. In *The 6th IEEE International Conference on Network Protocols (ICNP)*, Osaka, Japan, October, 2000.

[8] http://www.isi.edu/nsnam/ns/. *The Network Simulator - ns-2.*

[9] Y. J. Liang, E. G. Steinbach, and B. Girod. Multi-stream voice over ip using packet path diversity. In *the IEEE Fourth Workshop on Multimedia Signal Processing*, Cannes, France, October 2001.

[10] Y. J. Liang, E. G. Steinbach, and B. Girod. Real-time voice communication over the internet using packet path diversity. In *the ACM Multimedia Conference*, Ottawa, Canada, September/October 2001.

[11] N. F. Maxemchuk. Dispersity routing. In *the IEEE International Conference on Communications*, San Francisco, California, June 1975.

[12] N. F. Maxemchuk. Dispersity routing in high-speed networks. In *the Workshop on Very High Speed Networks*, Greenbelt, Maryland, March 1991.

[13] N. F. Maxemchuk. Dispersity routing in an atm network. In *the IEEE INFOCOM*, San Francisco, California, March/April 1993.

[14] P.M. Morse. *Queues, Inventories, and Maintenance.* John Wiley, 1958.

[15] T. Nguyen and A. Zakhor. Distributed video streaming with forward error correction. In *the International Packetvideo Workshop*, Pittsburg, Pennsylvania, April 2002.

[16] T. Nguyen and A. Zakhor. Distributed video streaming over internet. In *the SPIE Conference on Multimedia Computing and Networking*, San Jose, California, January 2002.

[17] S.M. Ross. *Stochastic Processes.* John Wiley, 1996.

[18] D. Rubenstein, J. Kurose, and D. Towsley. Detecting shared congestion of flows via end-to-end measurement. In *the ACM Sigmetrics Conference*, Santa Clara, California, June, 2002.

Appendix: Convergence of Data Loss Rate

In this appendix, we provide an explanation for the possible convergence of the data loss rate when the FEC group size is increased (e.g., by keeping the ratio of n/k but increasing the value of n). Let $P_{p-path}(j,n)$ be the probability of losing j packets under p parallel senders/paths when the FEC group size is n. Based on the derivation in Section 2, we have:

$$\begin{aligned} P_{1-path}(j,n) &= P^{(1)}(j,n) \qquad\qquad \text{for } j = 1,2,\ldots,n. \\ P_{m-path}(j,n) &= \sum_{i_1+\cdots+i_m=j} P^{(1)}(i_1,n_1) * P^{(2)}(i_2,n_2) * \cdots * P^{(m)}(i_m,n_m) \\ &\qquad\qquad \text{for } n_1+\cdots+n_m = n \text{ and } m > 1. \end{aligned}$$

Let $\Psi_{N-path}(n)$ be the average number of lost packets when we use $N \geq 1$ parallel senders and the FEC group size is n. We have that

$$\Psi_{N-path}(n) = \sum_{j=n-k+1}^{n} j P_{N-path}(j,n).$$

Let $\sigma = \frac{n-k}{n}$ be the fraction of redundant packets within a FEC group and P_{N-path} be the probability of losing any packet when one uses N parallel senders. We have that $P_{N-path} = \sum_{i=1}^{N} \alpha_i \frac{\mu_0(i)}{\mu_0(i)+\mu_1(i)}$. Let $\mathcal{L}_{p-path}(n,\sigma)$ denote the average data loss rate when a FEC group size of size n is used and σn is the number of redundant packets with $p \geq 1$ parallel senders. We conjecture that

$$\lim_{n\to\infty} \mathcal{L}_{p-path}(n,\sigma) = \begin{cases} 0 & \text{if } \lim_{n\to\infty} \frac{\Psi_{N-path(n)}}{\sigma n} \to 0, \\ (0,P_{N-path}] & \text{otherwise.} \end{cases} \tag{32}$$

The above statement is intuitive for the following reasons (its proof is left for future work). As we increase n (but keep σ constant), if the rate of increase of $\Psi_{N-path}(n)$ is less than the rate of increase of σn, then we will have more redundant packets to "protect" the lost packets within a FEC group; in that case, the average data loss rate $\mathcal{L}_{p-path}(n,\sigma)$ will converge to zero as we increase n. On the other hand, if the rate of increase of $\Psi_{N-path}(n)$ is greater than the rate of increase of σn, as we increase n, then we will have some irrecoverable packet losses within a FEC group. In that case, $\mathcal{L}_{N-path}(n,\sigma)$ has to be greater than zero and in the *worst* case, it is upper bounded by the packet loss rate of the channel.