

Picture-perfect Streaming Over the Internet: Is There Hope?

B. Abdouni* W. C. Cheng† A. L.H. Chow† L. Golubchik‡ J. C.S. Lui§
abdouni@usc.edu bill.cheng@usc.edu lhchow@cs.usc.edu leana@cs.usc.edu cslui@cse.cuhk.edu.hk

Abstract

Quality of service (QoS) in streaming of continuous media (CM) over the Internet is poor, which is partly due to variations in delays, bandwidth limitations, and packet losses. Although CM applications can tolerate some missing data, non-recoverable information loss degrades these applications' QoS. Consequently, a number of application areas (e.g., those related to the entertainment industry) have backed away from streaming of their content over the Internet. Inability to control the resulting visual and auditory quality of the resulting streamed presentation is an important reason for such a trend. We believe that this trend can be reversed. To this end, our work focuses on providing high quality streaming through the *exploitation of multiple paths* existing in the network. By high quality, we mean with significant bandwidth requirements, of relatively long duration, and without information loss or hiccups in data delivery. In this article, we present *evidence* that *multi-path streaming* is a promising approach.

Keywords: Multi-path streaming; application-level streaming over the Internet; quality-of-service.

1 What's Our Approach?

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, bandwidth limitations, as well as losses experienced by packets sent through wide-area networks. Although many such applications can continue to operate with some degree of missing data, non-recoverable information loss degrades these applications' quality of service. Consequently, a number of application areas (e.g., those related to the entertainment industry) have backed away from streaming of their content over the Internet. Inability to control the resulting visual and auditory quality of the resulting streamed presentation is an important reason for such a trend.

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 our work we investigate the potential benefits of providing QoS guarantees in continuous media delivery through the *exploitation of multiple paths* existing in the network between a sender (or a set of senders) and a receiver. The basic idea is that continuous media data (such as video) destined for a particular receiver is fragmented into packets and the different packets take alternate routes to the 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. There are many other advantages of multi-path (MP) streaming over the Internet — our focus in this article is to expose such benefits (in the context described below).

There are a number of approaches to accomplishing multi-path data delivery, and we describe the specific approach considered in this article 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 many potential benefits; we now describe several main ones.

Increased bandwidth. 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

*Dept. of Electrical Engineering Systems, University of Southern California, Los Angeles, California, USA.

†Dept. of Computer Science, University of Southern California, Los Angeles, California, USA.

‡Dept. of Computer Science, EE-Systems, IMSC, ISI, University of Southern California, Los Angeles, California, USA.

§Dept. of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.

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. Moreover, additional bandwidth can also allow for sending of a larger fraction of redundant information (e.g., in the form of erasure codes) which in turn also increases the resulting quality of delivered data.

Improved loss characteristics. 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. Sending data through multiple paths can potentially reduce burst lengths and correlations between consecutive losses and thus improve the quality of delivered data. We largely focus on this benefit in the remainder of the article.

Ability to adapt. 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 different parts of the network.

However, to reap the benefits of multi-path streaming one must first consider the following issues and solve the resulting research problems.

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 [3], what we need to be able to do is determine whether one path shares a point of congestion with another path [8, 15]. In addition, it may be useful (for some approaches to multi-path streaming) to be able to estimate current capacities and loss characteristics of these bottlenecks. Although this is not necessary in our approach, other approaches to multi-path streaming might require fairly accurate estimation of such network characteristics.

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 [2] 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 redistributing the load and sending less data on congested paths, and more on the less congested ones).

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, where it may be necessary to use a collection of relay hosts or proxies to “force” paths different from those provided by the network between a sender and a receiver. In contrast, this may be less of an issue for applications where data is pre-recorded and 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. We note that this is not a consideration in our approach, i.e., we require no support from the network.

Although all these issues are of importance, in this article 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 [15];
- 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 article we assume that the data is fully replicated at all servers and hence any server can deliver any fraction of the CM data.

An example of such a 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 $0 \leq \alpha_i \leq 1$

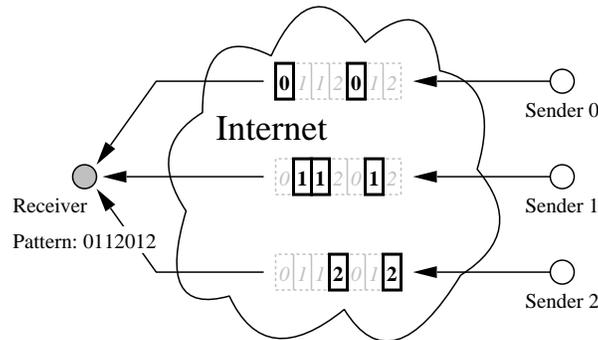


Figure 1: Continuous media system using multipath streaming.

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. (We elaborate on the remainder of the figure as well as on these issues later in the section 3) Thus, the receiver assembles the data from multiple senders and plays it in the appropriate order.

In this article, we present *evidence* that multi-path streaming is a promising approach to providing high quality streaming over the Internet. Although much research still remains on this topic, we believe that this evidence indicates that there is hope for being able to stream continuous media over the Internet, with high quality and without huge costs. Moreover, one can quickly deploy this type of service without requiring any modification to the core network infrastructure, which significantly simplifies the deployment process.

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 include dispersity routing, e.g., [11]. The focus in this work was on reducing delay, as opposed to our work where 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, partly because there is usually little opportunity to retransmit data in such real-time applications, and hence some amount of lossiness may have to be tolerated. We note that in a complete system a hybrid approach (with retransmissions and forward error correcting schemes) might be useful. However, this paper is focused on exploring loss correlations which have a more significant effect on error correcting schemes; we believe that adding retransmission to our evaluation process

would obscure the characteristics being illustrated in this article.

Another set of works on the topic considers higher level mechanisms, but requires 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, e.g., in [5, 4]. 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.

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. 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 –*an advantage being that it can then be made to work well with any coding technique or media codecs*. 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).

To the best of our knowledge, our work [7] is the first to provide an analytic framework for illustrating the benefits of MP streaming as compared to SP streaming, using metrics which describe both mean loss rates as well as correlations between losses (more details on the importance of these metrics are given below). However, much research on this topic remains to be done. This is evident by a number of recent works on this topic. For instance, in [13] the authors also consider 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 [12] FEC techniques are added (as compared to [13]) with the same goals in mind. In contrast, due to the nature of the application, we believe that it is also important to consider loss characteristics even when the losses cannot be fully recovered because video can be displayed under some losses. Hence, our work considers performance metrics such as data loss rate, error burst length distribution, as well as lag-1 autocorrelation (all with and without the use of erasure codes), in our evaluation of potential benefits of multi-path streaming. Another work [1] uses multiple description coding in addition to multiple paths, as opposed to using FEC, thereby tying the benefits of the technique to a particular coding scheme. In contrast, we believe that it is important to understand the benefits of MP streaming independently of data recovery schemes used. The applicability of this study (at least to future streaming applications and network conditions) is also somewhat limited in that (a) it constrains the bursts on paths to no more than 3 and (b) it only considers cases where the loss rates across multiple paths are fixed at 5%. In [14], the authors use relay nodes to provide multiple paths for delivery of streaming media (this is also suggested as a possibility above). To ensure diverse paths, their system requires a good deal of detailed information about the underlying network topology, e.g., router names and link properties. In contrast, the techniques described in Section 3 do not require such information.

Before proceeding with the exposition of benefits of MP streaming, we note that one should also consider the potential costs or detrimental effects of such an approach. For instance, MP streaming might have an adverse effect on the resulting delay characteristics observed at the receiver. As a result, it might require a large amount of receiver buffer space. Also, a simple implementation of MP streaming would require replicas of the CM data to be distributed to the servers (although, mirroring servers is a common practice). As pointed out above, advances in data placement approaches are needed to reduce such costs. 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 need to detect shared points of congestion. Other approaches to MP streaming might require even more detailed information about the network 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. Although, a quantitative evaluation of such costs is outside the scope of this paper, we discuss some of the costs-related issues in the context of our implementation (refer to Section 3).

2 What Are the Benefits?

In this section we focus on the exposition of MP streaming benefits in the context of losses, using the following metrics:

- **Loss rate:** the fraction of lost packets as seen by the receiver.
- **Lag-1 autocorrelation:** the degree of dependency of consecutive packet losses as seen by the receiver.
- **Conditional burst length of lost packets:** the probability mass function of consecutively lost packets as seen by the receiver.

We choose to consider the above metrics, as opposed to simply considering loss rates (i.e., only the first metrics), because we believe that the correlation among packet losses can significantly affect the resulting quality of the delivered media. Hence, we first describe a small experiment to illustrate that such correlations matter.

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 $i + 1, i + 2, \dots, i + k$ if frame i is received successfully and frames $i + 1, \dots, i + k$ are loss.

For each possible frame loss pattern, we measure the quality of the received video by computing the corresponding peak signal-to-noise ratio (PSNR). (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 value¹ 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 [6] 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 non-recoverable information loss (e.g., 2% in our experiment), the PSNR metric can be significantly lower for the more bursty loss patterns, and hence can 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.

Error Burst Length	Lost Frames Numbers	PSNR (dB)
1	25+k*50 where $k \in \{0, 1, \dots, 29\}$	39.107 dB
2	{50,51} + k*100 where $k \in \{0, 1, \dots, 14\}$	38.015 dB
3	{74,75,76} + k*150 where $k \in \{0, 1, \dots, 9\}$	31.325 dB
5	{123,124,....,127} + k*200 where $k \in \{0, 1, \dots, 5\}$	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 with a constant (2%) loss rate.

We now present results of a quantitative evaluation of loss characteristics of single path (SP) vs. multi-path (MP) streaming using simulations and based on the metrics described above. In our simulations, as in [3], we use a two-state Markov chain, known as the Gilbert model, to model a path’s loss characteristics. This is a commonly used model (in

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

the networking literature) of losses in wide-area networks such as the Internet. Specifically, the packet loss process along path i is described by a two state continuous time Markov chain $\{X^{(i)}(t)\}$ where $X^{(i)}(t) \in \{0, 1\}$. If a packet is transmitted at time t when the state of path i is $X^{(i)}(t) = 0$, then the transmitted packet is received correctly by the receiver. On the other hand, the transmitted packet is considered lost if $X^{(i)}(t) = 1$. The infinitesimal generator for the Gilbert model characterizing path i is:

$$Q(i) = \begin{bmatrix} -\mu_0(i) & \mu_0(i) \\ \mu_1(i) & -\mu_1(i) \end{bmatrix}$$

As in [3] we characterize each path by its bottleneck link. This model allows for dependence in consecutive packet losses and should be a more accurate representation of the loss process in the network than an independent loss model.

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, and these paths are assumed to be independent. The loss process of path i is modeled by a Gilbert model. Unless stated otherwise, we use $\mu_0(i) = 20$ and $\mu_1(i) = 70$, for $i = 1, 2, 3$. 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 experiments presented below are carried out with and without the use of erasure codecs. We perform experiments without the use of erasure codes in order to gain insight into the behavior of MP streaming, independently of the techniques used to recover information missing due to packet losses. Since it is typical to include redundant information in a streaming application (for the purpose of reconstructing information missing due to packet losses), experiments are also carried out with the use of FEC [2]. When FEC is added to a video stream, 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. When a packet cannot be reconstructed, we consider this as *non-recoverable information loss*.

Experiment (Data Loss Rate): Here we study the data packet loss rate of the SP and MP approaches using 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 (as expected).
- 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 (as derived in [7]), 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 increased bandwidth) are the important consideration. That is, when multiple paths are available (but the need for additional bandwidth 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 (refer to [7] for further comparison to a best-path type approach).

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.

Experiment (Conditional Error Burst Length): Here we compare the conditional burst length distribution, conditioned on there being at least one error. Figure 2(a) illustrates the conditional probability mass functions of error burst

length. We observe that the packet error burst length is indeed stochastically less than the error burst length of the single path streaming. This relationship also holds when we employ FEC.

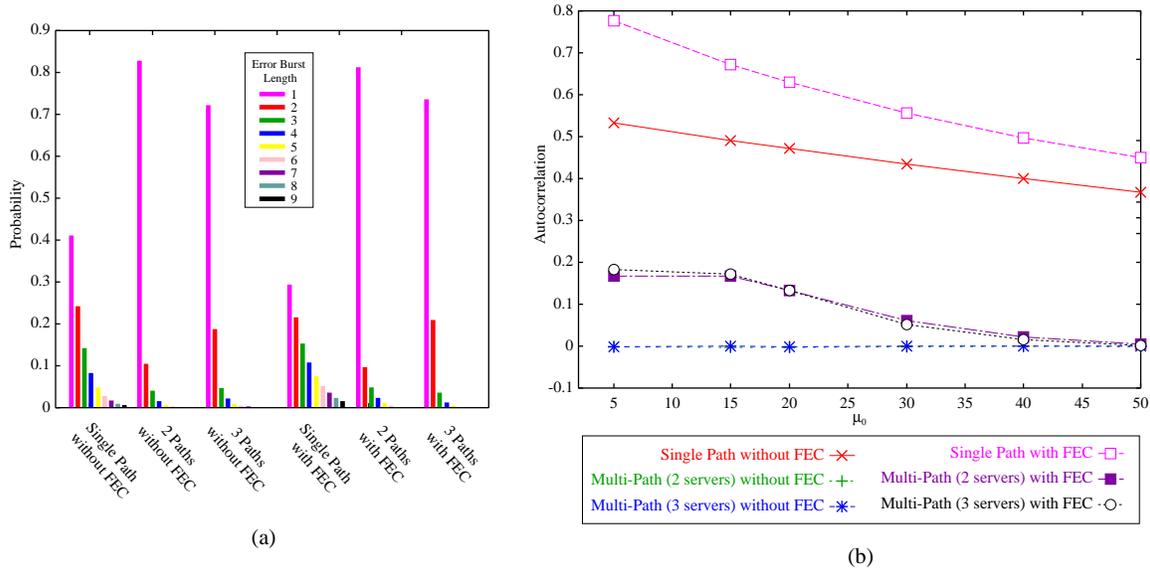


Figure 2: (a) Conditional probability mass functions of error burst length, (b) Lag-1 autocorrelation.

Experiment (Lag-1 Autocorrelation): Here we study the lag-1 autocorrelation of packet losses for both SP and MP streaming. Figure 2(b) illustrates the lag-1 autocorrelation where $\mu_0(i)$ 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 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.

Experiment (FEC Overhead): Here we study the effect of using FEC for both SP and MP streaming. Two homogeneous paths are used for MP streaming. Three scenarios where paths carry 0.5%, 5%, 10% loss rate are studied to find the minimum FEC overhead $(n - k)/k$ required to achieve a resulting quality of service, which in this case we define as non-recoverable information loss rate of $\leq 0.1\%$. We vary $\mu_0(i)$ accordingly to emulate the above given path loss rates. For each scenario, different FEC group sizes are tested by varying k . Table 3 gives the mean FEC overhead requirement, averaged over 10 experiments. These results are reported with 95% +/- 11% confidence intervals. From the results, we make the following observations.

- Given the requirement of non-recoverable information loss rate of $\leq 0.1\%$, MP streaming requires less FEC overhead than SP streaming. For example, when path loss rates are 0.5% and FEC group sizes are 8 SP streaming requires an overhead of 98.75% while MP streaming requires an overhead of only 37.50%.
- Given a FEC overhead constraint, MP streaming allows us to use a smaller FEC group size than SP streaming. For example, if the system can only afford a 20% overhead when path loss rates are 0.5%, then a FEC group size of 16 can be used in MP streaming while a group size of 32 is needed in SP streaming in order to achieve a non-recoverable information loss rate of $\leq 0.1\%$. In this case, MP streaming should result in shorter streaming latency and smaller buffer size requirements.

Path Loss Rate	k=8	k=16	k=32	k=64	k=128
0.5% (SP)	98.75%	40.63%	18.44%	9.22%	5.00%
0.5% (MP)	37.50%	18.75%	9.38%	4.69%	3.16%
5% (SP)	$\geq 1000\%$	147.50%	58.75%	31.88%	19.92%
5% (MP)	97.50%	55.63%	31.25%	20.31%	14.14%
10% (SP)	$\geq 1000\%$	218.13%	88.44%	50.00%	33.44%
10% (MP)	166.25%	85.00%	49.69%	34.06%	25.16%

Table 3: FEC overhead to achieve a non-recoverable information loss rate of $\leq 0.1\%$.

3 Can This be Done?

To verify our analytical and simulation results under real world conditions, we designed and built a multipath streaming system, currently being used for streaming video experiments between UMD, CUHK, and USC. Our prototype consists of a streaming module built on top of a P2P substrate. We chose to use a P2P network as the supporting architecture for our prototype because of the parallelism between simultaneous downloads in P2P systems and multipath streaming. In P2P network, a user can download parts of the same file from multiple peers, i.e., similarly to what we do with MP streaming.

3.1 P2P Architecture

The P2P network we built upon is a Gnutella-style network in that searches are flooded to all the nodes within a certain number of hops of the node sending the search query. Unlike Gnutella, the responses to search queries are not routed back through the P2P network, but are instead sent directly to the searching node on a dedicated connection. This takes some load off the P2P network, but it sacrifices the anonymity of the nodes involved in the search. The searching node proceeds to collect responses to its query, in much the same way as a Kazaa client does. After the search process is over, the user has a list of matching files, each with its own list of nodes hosting the file. The user then proceeds to select the file he/she wants to stream and the node from which he/she wants to stream. The file is then streamed from multiple nodes, i.e., to achieve MP streaming.

3.2 Control Channel

The streaming requests are sent directly to the senders over dedicated control channels. The receiver then proceeds to send control information to each sender. This information includes the file name, starting position, stream time, and synchronization information. The first three are basic, but the fourth is a somewhat tricky part. For each sender to be able to send a unique packet, they must know what the others are sending. Our design has connections from the receiver to the senders but not among the senders themselves. The only way to ensure that the senders are in-sync is to provide them with a sending order, or pattern, and a sender identifier. The receiver would in this case send the same sending pattern to all senders, while sending to each sender its unique identifier. The control channels to the senders are also responsible for changing the load on each path by changing the sending pattern.

For instance, suppose the receiver wants to stream from three senders as in Figure 1. Then, the receiver assigns a sender-ID to each sender and a starting point in the media file. The senders will then proceed to packetize the media file and match it to the sending pattern. However, they will only send packets when the ID in the sending pattern corresponds to the local sender's ID. This is depicted in Figure 1 where the receiver sends the pattern 0112012 to each sender along with their sender IDs. After packetizing the requested file, the senders proceed to check if it is their responsibility to send a particular packet. For example, Sender 0 will send the first packet, ignore the second, third, and fourth packets, and send the fifth packet, ignoring the sixth and seventh, after which the cycle begins again. Sender 1 on the other hand, will only send every second, third, and sixth packet of every group. Lastly, Sender 2 will send the fourth and the seventh packet of every group. Note that the sending pattern effectively dictates the load distributions on each sender/path.

3.3 Streaming Module

Given MP streaming parameters (i.e. media file, number of senders, sending pattern, and FEC parameters), the MP streaming module performs the basic streaming functions of reading data from the media source file, packetizing the data, encoding/decoding FEC-related information, transmitting packets, receiving them, and decoding the video. The streaming modules on the senders are synchronized by the aforementioned technique.

For added flexibility, the streaming module provides raw packet information, including time of arrival and sequence numbers. When added to the sending pattern and sender IDs, this data provides insight into the characteristics of each path, e.g., losses and their distributions, and the total performance of the system after merging the streams. These performance numbers can be used to tune the sending pattern, and therefore load distribution on senders/paths, for the best performance. These measurements can also be used to determine shared points of congestion in order to select appropriate senders/paths for MP streaming. In our prototype, we are currently implementing both passive and active techniques for detecting shared points of congestion, as proposed in [15, 8]. These measurements are also useful for validating our analytical and simulation results with data collected over the Internet. They can also be used as input to trace-driven simulations, as they represent real-world conditions.

4 Is it Worth the Trouble?

In this article we focused on multi-path streaming as an approach to providing high quality streaming over the Internet. We presented evidence that such an approach has promise as well as pointed out a number of open research problems in this context. Although much research still remains on this topic, we believe that this evidence indicates that there is hope for being able to stream continuous media over the Internet via the multi-path approach, with high quality and without huge costs.

5 Acknowledgment

This research has been funded in part by the NSF ANI-0070016, NSF EIA-0091474, RGC and Mainline Research grants and the Croucher Foundation Scholarship. It has also been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

References

- [1] John Apostolopoulos, Tina Wong, Susie Wee, and Dan Tan. On multiple description streaming with content delivery networks. In *The IEEE InfoCom*, New York, New York, June, 2002.
- [2] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley, January 1983.
- [3] Jean-Chrysostome Bolot, Sacha Fosse-Parisis, and Don Towsley. Adaptive FEC-Based Error Control for Internet Telephony. In *INFOCOM*, 1999.
- [4] 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.
- [5] H. Chu and K. Nahrstedt. Dynamic multi-path communication for video traffic. In *the Hawaiian International Conference on System Science*, Hawaii, January, 1997.
- [6] D. Le Gall. MPEG: a Video Compression Standard for Multimedia Applications. *Communications of the ACM*, April 1991.
- [7] L. Golubchik, J. C.S. Lui, T. F. Tung, A. L.H. Chow, W.-J. Lee, G. Franceschinis, and C. Anglano. Multi-path Continuous Media Streaming: What are the Benefits? *Performance Evaluation*, 49:429–449, September 2002.
- [8] Khaled Harfoush, Azer Bestavros, 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.
- [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] T. Nguyen and A. Zakhor. Distributed video streaming with forward error correction. In *the International Packetvideo Workshop*, Pittsburg, Pennsylvania, April 2002.
- [13] T. Nguyen and A. Zakhor. Distributed video streaming over internet. In *the SPIE Conference on Multimedia Computing and Networking*, San Jose, California, January 2002.

- [14] Tinh Nguyen and Avidah Zakhor. Path diversity with forward error correction (pdf) system for delay sensitive applications over the internet. In *The IEEE InfoCom*, San Francisco, California, March, 2003.
- [15] Dan Rubenstein, Jim Kurose, and Don Towsley. Detecting shared congestion of flows via end-to-end measurement. In *IEEE/ACM Transactions on Networking*, volume 10, June 2002.

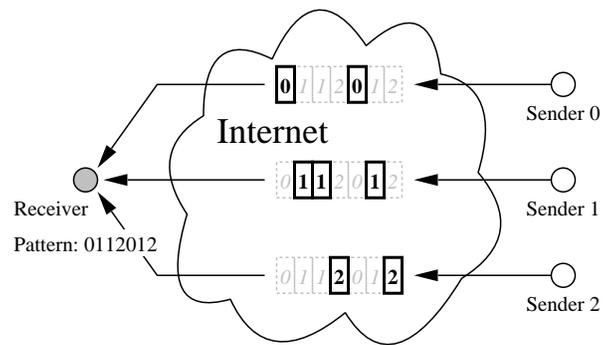
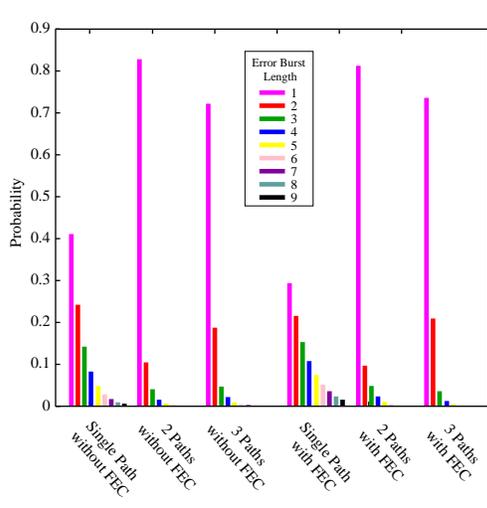
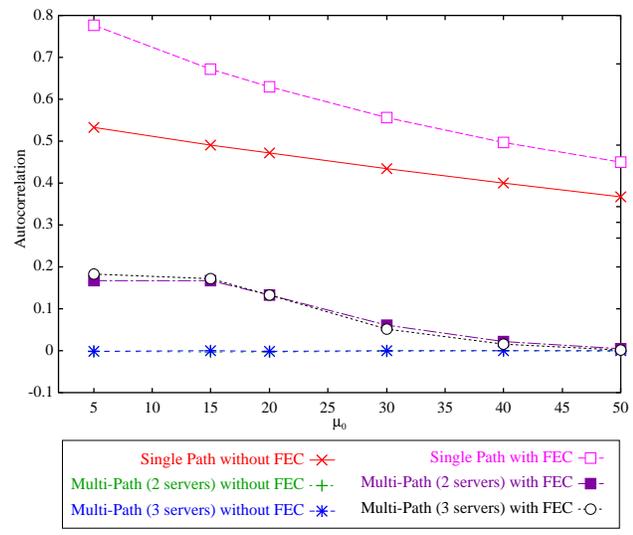


Figure 3: Continuous media system using multipath streaming.



(a)



(b)

Figure 4: (a) Conditional probability mass functions of error burst length, (b) Lag-1 autocorrelation.