PROTOCOLS FOR DISTRIBUTED VIDEO STREAMING

Thinh Nguyen and Avideh Zakhor

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
{thinhq, avz}@eecs.berkeley.edu

ABSTRACT

With the explosive growth of video applications over the packet switched networks, many approaches have been proposed to stream video effectively over packet switched, best-effort networks. In our previous work, we proposed a framework with a receiver driven protocol to coordinate simultaneous video streaming from multiple senders to a single receiver in order to achieve higher throughput, and to increase tolerance to packet loss and delay due to network congestion. The receiver-driven protocol employs two algorithms: the rate allocation and packet partition. The rate allocation algorithm determines the sending rate for each sender, while the packet partition algorithm ensures no sender sends the same packets, and at the same time, minimizes the probability of late packets. In this paper, we extend the rate allocation scheme to be used with Forward Error Correction (FEC) in order to minimize the probability of packet loss in a bursty loss environment such as one due to network congestion. Using both simulations and actual Internet experiments, we demonstrate the effectiveness of our rate allocation scheme in reducing packet loss, and hence, achieving higher visual quality for the streamed video.

1. INTRODUCTION

Video streaming over best-effort, packet-switched networks is challenging due to a number of factors such as, high bit rates, delay, and loss sensitivity. To this end, many solutions have been proposed. From source coding perspective, layered and error-resilient video codecs have been proposed to deal with packet loss, heterogeneity, and time-varying nature of the Internet [2]. From network perspective, there are approaches based on multicast [6] and TCP-friendly protocols [2, 3] to reduce bandwidth and jitter for streaming multimedia data over the Internet. A number of these schemes assume a single fixed route between the receiver and the sender throughout the session. If the network is congested along that route, video streaming suffers from high loss rate and jitter. Based on these, our previous work [1] proposed a distributed video streaming framework in which multiple senders simultaneously stream video to a single receiver to effectively provide the required throughput as shown in Figure 1. Having multiple senders is also a diversification scheme in that it combats unpredictability of congestion in the Internet. If the route between a particular sender and the receiver experiences congestion during streaming, the receiver can redistribute streaming rates among other senders, thus resulting in smooth video delivery. There have been other works dealing with simultaneous downloading of non-real time data from multiple mirror sites. For example, the authors in [5] use Tornado codes to download data simultaneously from multiple mirror sites.

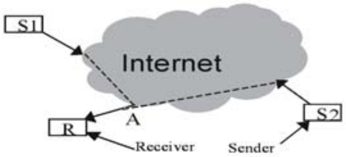


Fig. 1. *Distributed video streaming architecture*

To successfully stream video from multiple senders, we assume that the available aggregate bandwidth from all the senders to the receiver exceeds the required video bit rate. We also assume that the routes from the client to the senders do not share the same congestion link. If there is congestion on a shared link between two senders, the lost packets between different senders are correlated, our assumption of independent packet losses between routes for the analyses on optimal sendingrates will no longer hold.

The previously proposed rate allocation algorithm in [1] aims to minimize number of lost packets, assuming the lost packets are identically and independently distributed. However, this rate allocation algorithm is based on a packet loss model with uniform distribution, which has been shown to be inadequate for bursty packet loss due to congestions in the packet switched networks. In addition, our previous rate allocation does not use FEC to protect data, potentially leading to unacceptably long delay due to retransmissions of lost packets. In this paper, we extend rate allocation scheme in [1] to be used with FEC in bursty loss environments. The rest of our paper is organized as follows. In Section 2, we briefly describe our previously proposed transport protocol. In Section 3, we introduce a novel rate allocation scheme to be used with FEC minimizing the probability of packet loss. In Section 4, we present simulations and actual Internet experimental results. Finally, we will conclude in Section 5.

2. PROTOCOL OVERVIEW

2.1. Transport Protocol

In this section, we briefly describe our protocol originally proposed in [1]. Our transport protocol is a receiver-driven one in which, the receiver coordinates transmissions from multiple senders based on the information received from the senders. Each sender estimates and sends its round trip time to the receiver. The receiver uses the estimated round trip times and its estimates of senders' loss rates to calculate the optimal sending rate for each sender. When the receiver decides to change any of the senders' sending rates due to changes in conditions of various routes, it sends an identical control packet to each sender. The control packet contains the synchronization sequence number and the optimal sending rates as calculated by the receiver for all senders. Using the specified sending rates and synchronization sequence number, each sender runs a distributed packet partition algorithm to determine the next packet to be sent.

2.2. Rate Allocation Algorithm

In our proposed protocol in [1], the receiver computes the optimal sending rate for each sender based on its loss rate and estimated available bandwidth as proposed in [3]. The problem of allocating optimal sending rate to each sender can be stated as follows. Let N be the total number of senders, and $L(i,t)$ and $S(i,t)$ be the estimated loss and sending rates, respectively for sender i over an interval $(t, t+\delta)$. Our goal is to find $S(i,t), i = \{1...N\}$, in such a way as to minimize the total lost packets during interval $(t, t+\delta)$ given by

$$F(t) = \sum_{i=1}^{N} L(i,t)S(i,t)$$

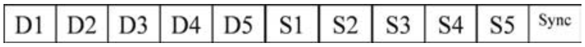
subject to $0 \leq S(i,t) \leq B(i,t)$ and $\sum_{i=1}^{N} S(i,t) = S_{req}(t)$, where S_{req} is the required bit rate for the encoded video during the interval $(t, t+\delta)$, and $B(i,t)$ is the TCP-friendly estimated bandwidth for sender i during the interval $(t, t+\delta)$. In [1], we

This work was supported by NSF under grant CCR-9979442 and AFOSR contract F49620-00-1-0327.

have proposed an algorithm to minimize $F(t)$, the number of lost packets during interval $(t, t+\delta)$, given instantaneous feedback, and assuming that the estimated loss rate and TCP-friendly available bandwidth are accurate. The idea of the algorithm is as follows. At time t, we sort the senders according to their estimated loss rates from lowest to highest. We start with the lowest loss rate sender and assign its sending rate to be its TCP friendly estimated bandwidth as described in [3]. We then continue to set each sender's rate to its available bandwidth, beginning with the ones with lower loss rates and moving to the ones with higher loss rates, until the sum of their available bandwidths exceeds the bit rate of the encoded video.

2.3. Packet Partition Algorithm

After receiving the control packet from the receiver, each sender immediately decides the next packet in the video stream to be sent, using the packet partition algorithm. All the senders simultaneously run this algorithm to ensure that no two or more senders send the same video packet, and also to minimize the probability of packets arriving late at the receiver due to network jitter. The algorithm can be described as follows. Each sender receives control packet from the receiver through a reliable protocol whenever the receiver determines there should be a change in any of the sending rates. The format of the control packet is shown Figure 2. $S1-S5$ are two-byte fields to specify the sending rate in packets/second for each sender. Packet size is constant for all senders. $D1-D5$ are one-byte fields to denote the estimated delay from each sender to the receiver. This delay is expressed in multiples of 2 milliseconds interval. The $Sync$ field is the starting sequence number that all senders use in the packet partition algorithm to determine the next packet to send, immediately upon receiving the control packet.



D1	D2	D3	D4	D5	S1	S2	S3	S4	S5	Sync

Fig. 2. *Packet format*

The basic idea in our packet partition algorithm is that among all senders $i = \{1...N\}$, the one that maximizes the time difference $A(i,k)$ between the estimated receive and playback time for k^{th} packet is chosen to send that packet. Hence, maximizing $A(i,k)$ is equivalent to minimizing the probability that the k^{th} packet is late. The details on estimating $A(i,k)$ and other practical issues can be found in [1].

3. PROPOSED DISTRIBUTED STREAMING WITH FEC

In this section, we extend our previous work by proposing a novel rate allocation scheme to be used with FEC to minimize the probability of packet loss in bursty loss environments. In general, FEC has been shown to be an effective tool in combatting packet loss in streaming applications on the packet switched networks [4]. The main drawback of FEC though is that it results in bandwidth expansion and hence reduces the amount of available bandwidth for the actual video bit stream. Since the level and burstiness of packet loss in the Internet fluctuates significantly, incorporating the optimal amount of FEC in any streaming application is a difficult task; too little redundancy cannot effectively protect the video bit stream, and too much redundancy consumes too much bandwidth unnecessarily. Thus FEC level has to be closely matched to channel characteristics for it to be effective in single route streaming applications. In this paper, we show that by combining path diversification and FEC, we can combat bursty loss behavior in the Internet more effectively. Specifically the above mismatch of FEC level and network characeteristics for single route streaming application becomes more relaxed in distributed streaming applications due to the additional redundancy introduced by multiple routes. We now provide a brief discussion on the network model.

3.1. Network Model

An accurate model for packet loss over the Internet is quite complex. Instead, we model our network as a simple two-state continuous-time Markov chain, which has been shown to approximate the behavior of packet loss over the Internet fairly well. A two-state continuous Markov chain with the state at time t denoted by X_t where $X_t \in \{g, b\}$, is characterized by μ_g and μ_b. μ_g and μ_b can be thought of as the rates at which the chain changes from the "good" state to the "bad" state and vice versa. When the chain is in the good state, the probability of having lost packets is much smaller than that of when the chain is in the bad state. A further simplied model assumes that a packet transmitted at time t is successfully received if $X_t = g$, and is lost otherwise.

3.2. Optimal Rate Allocation

To simplify analysis, we replace the two-state continuous-time Markov chain with an equivalent two-state discrete one. The transition probabilities for the discrete Markov chain is computed as follows:

$$p_{gg} \triangleq P(X_{n+1} = g | X_n = g) \quad = \quad \pi_g + \pi_b e^{-(\mu_g + \mu_b)\tau} \quad (2)$$

$$p_{gb} \triangleq P(X_{n+1} = g | X_n = b) \quad = \quad 1 - p_{gg}(\tau) \quad (3)$$

$$p_{bb} \triangleq P(X_{n+1} = b | X_n = b) \quad = \quad \pi_b + \pi_g e^{-(\mu_g + \mu_b)\tau} \quad (4)$$

$$p_{bg} \triangleq P(X_{n+1} = b | X_n = g) \quad = \quad 1 - p_{bb}(\tau) \quad (5)$$

where τ is the sending interval between packets. With the discrete model, the process of the discrete Markov chain undergoing n discrete time steps is equivalent to the process of sending n packets through the network. To further simplify analysis, we only consider the case of two senders, both assumed to be sending packets to the receiver along two routes with independent packet loss. The extension of analysis to the case with more than two senders is straightforward. Our goal is to find the sending rates for the two senders in order to (a) minimize the probability of irrecoverable loss for a given protection level of FEC, and (b) to ensure that each sender sends packets at a TCP-friendly rate. To formally state our rate allocation problem, we use the following notation:

S_{req} Required video sending rate in packets per second

N Total number of packets in FEC block.

K Number of data packets in FEC block.

B_m Estimated TCP-friendly bandwidth for sender m in packets per second

(μ_g^m, μ_b^m) Network parameters

$\lambda = \frac{N}{S_{req}}$ Interval between successive transmitted FEC block in seconds

N_m Number of packets transmitted by sender m during λ seconds

The rate allocation problem can now be stated as follows:
Given S_{req}, N, K, B_m, (μ_g^m, μ_b^m), we want to find N_m for $m = 0, 1$ so as to minimize the probability of irrecoverable loss given by

$$C(K, N_0, N_1) = \sum_{j=N-K+1}^{N_0+N_1} \sum_{i=0}^{j} P(0, i, N_0) P(1, j-i, N_1) \quad (6)$$

subject to

$$N_0 + N_1 = N, \quad \tfrac{N_0}{\lambda} \leq B_0, \quad \tfrac{N_1}{\lambda} \leq B_1 \quad (7)$$

where $P(m, i, N_m)$ is the probability that i packets are lost out of the N_m packets sent by sender m. $C(K, N_0, N_1)$ is the probability that more than $N-K$ packets are lost out of a total $N_0 + N_1$ packets sent by both senders. Since we assume independent packet loss along the two routes, the probability of j lost packets out of $N_0 + N_1$ packets sent by both senders can be written as $\sum_{i=0}^{j} P(0, i, N-0) P(1, j-i, N_1)$. Therefore, the probability of more than $N-K$ lost packets out of $N_0 + N_1$ packets sent is $\sum_{j=N-K+1}^{N_0+N_1} \sum_{i=0}^{j} P(0, i, N_0) P(1, j-i, N_1)$. As indicated in the constraints in (7), $\frac{N_m}{\lambda}$is the sending rate of sender m, which is required to be less than or equal to the estimated TCP-friendly bandwidth. Since the sum of the sending rates by the two senders equals to the required sending rate for the video, we have $N_1 + N_0 = N$. In this paper, we assume that the aggregate TCP-friendly rate is always greater than or equal to the video bit rate. The procedure to compute $P(m, i, N_m)$ is shown in the Appendix. Using $P(m, i, N_m)$, we search over all possible values of N_0 and N_1 such that the constraints in (7) are satisfied, and $C(K, N_0, N_1)$, the probability of irrecoverable packet loss is minimized.

3.3. Numerical Characterization

To compare the capability to recover lost packets using our optimal sending rate allocation for two senders against that of using one sender, we numerically compute and compare the probabilities of irrecoverable loss across different model parameters (μ_{good}, μ_{bad}) for the two following scenarios. In "two senders" scenario, packets are simultaneously sent along two "loss-independent" routes A and B while in "one sender" scenario, all packets are sent along route A. We vary the model parameters in our computations to determine the robustness of the optimal sending rate allocation under different network conditions. For convenience, we refer to the average time that a sender spends in "good" and "bad" states as average good and bad times, respectively. The parameters of the two routes vary as follows. The average good time of routes A and B are identical and they vary from $1s$ to $5s$. The average bad time of route A remains constant at $0.02s$ while average bad time of route B varies from $0.02s$ to $0.2s$. The probability that a packet is lost while in "good" and "bad" states is 0 and 1, respectively. The aggregate sending rate of both "two senders" and "one sender" scenario is $500kbps$, and the packet size is set to 500 bytes. In both scenarios, packets are protected using $RS(100, 88)$ codes.

Figure 3(a) shows the probability of irrecoverable loss for for the two sender scenario using our rate allocation algorithm in Section 3.2. The y-axis shows the average good times for both routes A and B ranging from $1s$ to $5s$, while the x-axis shows the average bad time of route B ranging from $0.02s$ to $0.2s$. The z-axis in the same figure shows the probability of irrecoverable loss using optimal rate partition between two routes A and B at different average good and bad times. As an example, the point $(0.05, 2, 0.01)$ in the graph indicates that the minimum probability of irrecoverable loss is 0.01 when the average good times for both routes A and B are $2s$, while the average bad time for routes A and B are $0.02s$ and $0.05s$, respectively. Figure 3(a) indicates that the probability of irrecoverable loss varies mostly with the average bad time while it remains relatively constant with respect to the average good time except when the average good time is small. This observation is intuitively plausible since we would expect a route with long average bad time to have longer bursts of lost packets, leading to higher probability of irrecoverable loss. The z-axis in Figure 3(b) shows N_A, the optimal number of packets out of 100 that should be sent on route A, with the remaining $100 - N_A$ packets sent on route B. This graph shows that as the average bad time of route B increases from $0.02s$ to $0.2s$, more packets should be sent on route A. This result is intuitively plausible since the average bad time of channel A is only $0.02s$, and therefore sending more packets on route A will result in smaller probability of irrecoverable loss. Even though route A has a lower average bad time than that of route B, sending all packets in the route A, is not always a good idea as shown Figure 3(c). Figure 3(c) shows the ratio of irrecoverable loss probabilities between the case when all packets are sent on route A and the case when the optimal rate allocation is employed to send packets on both routes A and B. When the average bad time of route B is greater than $0.07s$, the performances of the multi-sender and uni-sender schemes are almost identical as the ratio of irrecoverable loss probabilities is approximately 1. If however, the average bad time of route B is less than $0.07s$, it is advantagous to use both routes A and B to send packets at the appropriate rates. For certain model parameters, the irrecoverable loss probability using optimal rate partition scheme is almost 35 times less than that of the uni-sender one.

4. SIMULATIONS RESULTS

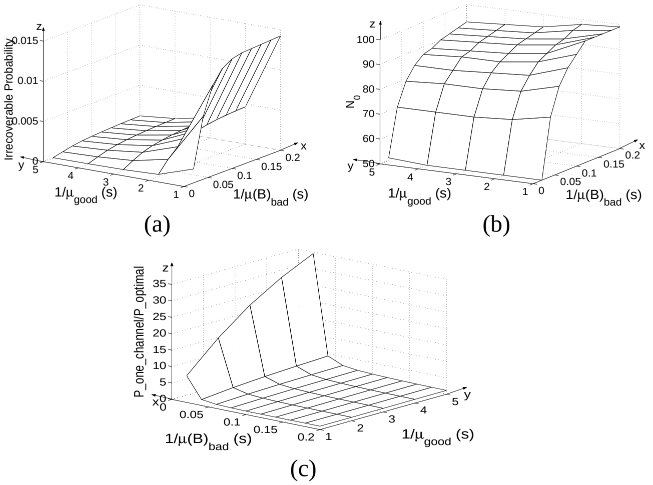
In this section, we perform both MatLab simulations and actual Internet experiments to show that our optimal rate allocation scheme results in fewer lost packets and leads to higher visual quality for the streamed video.

4.1. MatLab Simulations

To validate our numerical results of the optimal rate allocation, we perform the following two simulations. In the first simulation, we simulate a single sender and receiver by sending all the video packets using a single route in which, the packet loss behavior is modeled as a two-state continuous Markov chain. In the second simulation, we simulate multiple senders and single receiver by sending the video packets along two independent, identical routes similiar to the one in the first simulation. The simulation parameters are shown below

$\{\mu_{good}, \mu_{bad}\}$ = $\{0.15$**,** $30\}$
$S_{req} = 800kbps$**,** $720kbps$ video bit rate + $80kbps$ FEC rate
$packet\ size = 500$ bytes
$N = 100$ total number of packets in FEC block
$K = 90$ number of data packets in FEC block

The video in the simulations is a $1.5Mbps$ MPEG-1 video sequence taken from MPEG-7 test suite, which is then transcoded using H.263 encoder with error-resilient option at $720kbps$. The video is then packetized into $500bytes$ packets, and all packets are further protected by $RS(100, 90)$ codes, making the total video bit rate approximately $800kbps$. After FEC decoding at the receiver, if there is still an irrecoverable loss, we use a simple error-concealment technique to conceal the visually degraded frames. Basically, the error-concealment technique replaces the lost group of blocks (GOB) of the current frame with GOB of the previous frame, and copies the motion vectors of the lost GOB from the GOB above it. Using optimal rate allocation for two independent, identical routes with the above simulation parameters, we obtain the equal sending rates of $400kbps$ for each route. Figure 4(a) shows the number of lost packets per 100 versus packet sequence number in the first simulation where packets are sent at $800kbps$ using only one route. A point above the horizontal line represents an irrecoverable loss event. As seen, there are 7 instances of irrecoverable loss in which the number of lost packets exceeds 10. Figure 4(b) shows the number of lost packets out of 100 versus packet sequence number in the second simulation where packets are sent simultaneously at the optimal rate of $400kbps$ on each route. In this simulation, there is only one instance where FEC cannot recover the lost packets. Clearly, using the optimal sending rate allocation scheme to send packets on two "loss independent" routes results in fewer lost packets than that of sending all packets on only one route.



(a)

(b)

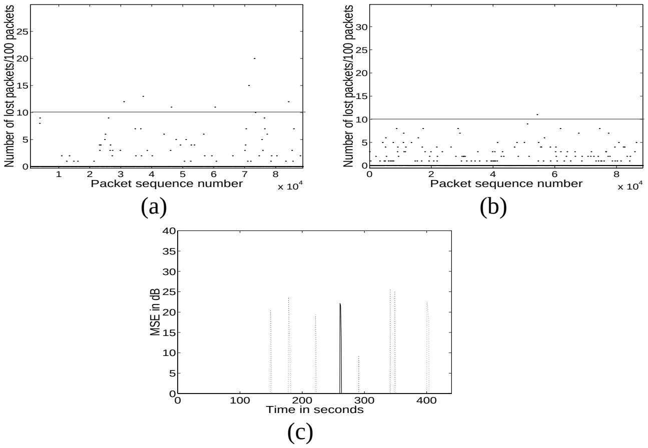
(c)

Fig. 3. *(a) Probability of irrecoverable loss for using optimal partition for two senders; (b) Optimal partition for for two sender; (c) Ratio of irrecoverable loss probability using only one sender to that of optimal scheme.*

Next, we compare the mean squared error (MSE) of pixel values between the sent and the received frames as a function of time. Higher MSE represents lower fidelity of the video. Figure 4(c) shows the MSE in dB, resulting from both experiments, with the dotted and solid lines representing the MSE from uni-sender and multi-sender simulations, respectively. As seen, peaks in MSE closely reflect instances at which irrecoverable loss occur. Visual inspection has shown that these peaks in MSE result in noticeable visual degradation of the video. Hence, the multi-sender scheme results in better visual quality than uni-sender scheme for the streamed video.

4.2. Internet Experiments

In addition to MatLab simulations, we have developed an actual system for distributed video streaming with real-time FEC decod-



(a)

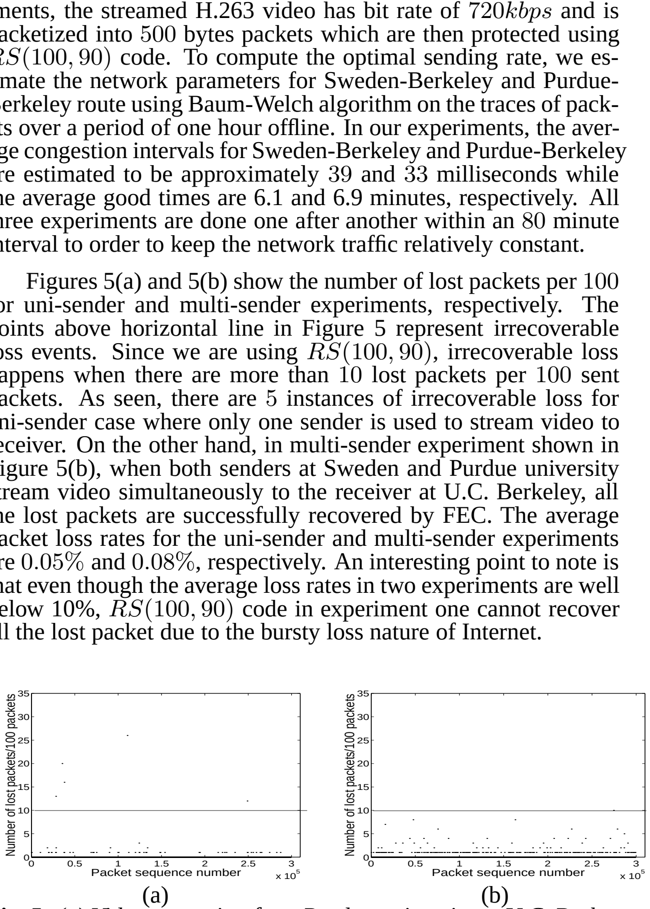
(b)

(c)

Fig. 4. *(a) Number of lost packets per 100 packets using one route; (b) Number of lost packets per 100 packets using two routes; (c) MSE from two simulations*

ing and displaying capabilities. We now demonstrate the effectiveness of the optimal rate allocation scheme in reducing the number of lost packets by performing the following two Internet streaming experiments. In experiment one, a sender at Purdue university streams a H.263 video to a receiver at U.C. Berkeley at the rate of 200 packets per second. In experiment two, both senders at Sweden and Purdue university simultaneously stream the video to a receiver at U.C. Berkeley with the optimal rates of 80 packets per second and 120 packets per second, respectively. In both experiments, the streamed H.263 video has bit rate of $720kbps$ and is packetized into 500 bytes packets which are then protected using $RS(100, 90)$ code. To compute the optimal sending rate, we estimate the network parameters for Sweden-Berkeley and Purdue-Berkeley route using Baum-Welch algorithm on the traces of packets over a period of one hour offline. In our experiments, the average congestion intervals for Sweden-Berkeley and Purdue-Berkeley are estimated to be approximately 39 and 33 milliseconds while the average good times are 6.1 and 6.9 minutes, respectively. All three experiments are done one after another within an 80 minute interval to order to keep the network traffic relatively constant.

Figures 5(a) and 5(b) show the number of lost packets per 100 for uni-sender and multi-sender experiments, respectively. The points above horizontal line in Figure 5 represent irrecoverable loss events. Since we are using $RS(100, 90)$, irrecoverable loss happens when there are more than 10 lost packets per 100 sent packets. As seen, there are 5 instances of irrecoverable loss for uni-sender case where only one sender is used to stream video to receiver. On the other hand, in multi-sender experiment shown in Figure 5(b), when both senders at Sweden and Purdue university stream video simultaneously to the receiver at U.C. Berkeley, all the lost packets are successfully recovered by FEC. The average packet loss rates for the uni-sender and multi-sender experiments are 0.05% and 0.08%, respectively. An interesting point to note is that even though the average loss rates in two experiments are well below 10%, $RS(100, 90)$ code in experiment one cannot recover all the lost packet due to the bursty loss nature of Internet.



(a)

(b)

Fig. 5. *(a) Video streaming from Purdue university to U.C. Berkeley; (b)Simultaneous video streaming from Sweden and Purdue university to U.C. Berkeley*

5. CONCLUSIONS

In this paper, we have proposed a novel rate allocation algorithm to be used with FEC in order to minimize the probability of lost packets in a bursty loss environment due to the network congestion. Using both MatLab simulations and actual Internet experiments, we demonstrate the effectiveness of our rate allocation scheme in reducing packet losses, and hence, achieving higher visual quality for the streamed video.

6. REFERENCES

[1] T. Nguyen and A. Zakhor, “Distributed Video Streaming”, Multimedia Computing and Networking 2002, Multimedia Conference, Jan 2002

[2] W. Tan and A. Zakhor, “Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol,” IEEE Transactions on Multimedia, vol. 1, p.172-86, June 1999

[3] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” Applications, Technologies, Architectures and Protocols for Computer Communication, p.43-56, Oct. 2000.

[4] J. Bolot, S. Fosse-Parisis, D. Towsley, “Adaptive FEC-based error Control for Internet Telephony”. IEEE INFOCOM ’99.

[5] J. Byers, M. Luby, and M. Mitzenmacher, “Accessing multiple mirror sites in parallel: using Tornado codes to speed up downloads,” Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, p.275-83, Mar. 1999.

[6] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Weiand, “The PIM architecture for wide-area multicast routing,” IEEE/ACM Transactions on Networking, vol. 4, p.153-62, April 1996.

APPENDIX
Procedure for computing $P(m, k, N_m)$

To compute $C(K, N_0, N_1)$ in Section 3.2, we first compute $P(m, i, N_m)$ based on the given network parameters (μ_g^m, μ_b^m) of sender m as follows. We denote

$S_m(n) \in \{g, b\}$ State of sender m after it sends n packets
$L_m(n)$ Number of lost packets out of n packets sent by sender m

$P_m^{loss}(i)$ Packet loss probability when sender m is in state i
p_{ij}^m Transition probability from state i to state j for sender m

Let $\phi_{ij}^m(k, n) \triangleq Prob(L_m(n) = k, S_m(n) = j | S_m(0) = i)$ denote the probability that sender m is in state j, and there are k lost packets after it sends n packets, given that it is intially in state i. We can compute $\phi_{ij}^m(k, n)$ recursively by conditioning on the previous state l, and by using the total probability theorem to obtain

$$\begin{aligned} \phi_{ij}^m(k, n) &= \sum_{l \in g,b} [\phi_{il}^m(k-1, n-1) p_{lj}^m P_m^{loss}(j) \\ &+ \phi_{il}^m(k, n-1) p_{lj}^m (1 - P_m^{loss}(j))] \end{aligned} \quad (8)$$

for all $k \geq 0$ and $n \geq 0$, with the boundary conditions:

$$\phi_{ij}^m(0, 0) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\phi_{ij}^m(k, n) = 0 \text{ for } n < k$$

Now, since $P(m, k, N_m)$ is the probability of k lost packets out of N_m packets sent by sender m, regardless of the initial and final states, we marginalize $\phi_{i,j}^m(k, N_m)$ to obtain

$$P(m, k, N_m) = \sum_{i \in \{g,b\}} \sum_{j \in \{g,b\}} \pi_i^m \phi_{ij}^m(k, N_m)$$

where $\pi_g^m = \mu_b^m / (\mu_g^m + \mu_b^m)$ and $\pi_b^m = \mu_g^m / (\mu_g^m + \mu_b^m)$ are the steady-state probabilities of sender m being in “good” and “bad” states, respectively.