Integrated Buffer Management and Congestion Control for Video Streaming

Anand Balan, Omesh Tickoo, Ivan Bajic, Shivkumar Kalyanaraman, John Woods

***Abstract*—In this paper we address the issue of efficient multimedia streaming by integrating an intelligent buffer management scheme with the congestion control scheme at the source. The scheme exploits the fact that most of the transmission losses actually occur at the *source* and not in the network. An intelligent transmission scheme can take advantage of this fact and thus control exactly what data is dropped in response to network congestion. The integrated model uses priority information from the encoder and network information from the congestion scheme and drops low priority packets and sends the most important packets in the available bandwidth. The packets are dropped when the source transmission buffer length exceeds a minimum threshold. This scheme ensures that the media transmitted has the highest possible quality under the given network conditions using a given coding scheme. This paper also presents a randomized transmission scheme as part of the integrated model to reduce the jitter and burst losses in the multimedia transmissions.**

***Index Terms*—Buffer management, multimedia networking, robust video streaming, frame rate scalable video.**

I. Introduction

MULTIMEDIA streaming over the Internet has attracted a lot of research interest in recent years. Most of the efforts so far have concentrated around developing and analyzing the various congestion control schemes and receiver side adaptation that can help reduce jitter for the multimedia traffic. The focal point of research in the multimedia streaming domain has been the development of an integrated video coding and congestion control approach to provide jitter free transmission of the media and fairness to the competing flows. Results have been published for interaction between the adaptive codecs and different transport schemes [2]. This paper, besides supplementing the the results reported already by other authors, shows that an order of magnitude performance improvement can be achieved by doing intelligent loss control at the source in response to congestion.

In this paper we present a new integrated scheme based on the inter-working between live adaptive encoding, differential packet filtering at the sender and TCP-friendly binomial schemes [3] that achieves lower loss, high quality and low jitter when multimedia is transported over the Internet.

The major hurdles to the effective multimedia transmission can be grouped into following:

- **Loss of important data in the network**. Most video encoding schemes encode video into packets with different importance and the packets are dropped in the transmission randomly. So the major hindrance in effective multimedia streaming is the loss of important data in the network.
- **Jitter**. Video also suffers from jitter due to the variation in rate of the congestion control scheme.
- **Burst losses** in network that lead to losing a set of packets containing information about a single frame making estimation techniques at the receiver ineffective.
- **Loss of synchronization** between the encoder and decoder due to network losses.
- **Loss of a significant amount of data with a loss of single packet** that renders quality reconstruction almost impossible.

In this paper we present an integrated solution to the above problems. The solution combines an *intelligent source buffer management* scheme with *binomial congestion control* schemes that are TCP friendly and have smoother rate variations. The buffer management scheme presented drops packets having lesser information content in priority to those having more important content whenever the network encounters congestion and needs the media flow to reduce its transmission rate. The problem arising due to burst losses is solved by introducing *"randomness"* in the transmision scheme as explained in later sections. We use a *robust codec* in our integrated approach to solve the problem of loss of synchronization due to isolated losses. The packetization scheme used in the integrated model does *dispersive packetization* that allows reconstruction using estimation and other techniques inpresence of a lossy network.

The rest of the paper is organized as follows: Subsection II-A presents an overview of the model we propose including the source buffer management algorithm. Subsection II-C motivates the need for robust scalable codecs and gives an overview of the robust motion compensated 3-D sub-band video coder we use in our model to report simulation results and we depict the generalized nature of the model across various types of codecs. Subsection II-D introduces the concept of binomial schemes and advantages of randomizing such schemes for multimedia transmission. Section III puts all the pieces together and we present an integrated *ns-2* simulation model. The simulation results are presented in section IV and we conclude with the main message of this paper in section V.

II. The Integrated Model

A. General Model

Video streaming requires a transmission scheme that gives a steady rate while reducing burst losses. Also the video encoders

Authors are with the Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA.

This work was supported in part by the ARO grant number 19-00-1-0559.

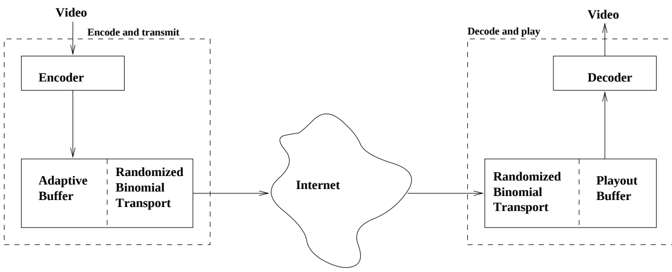
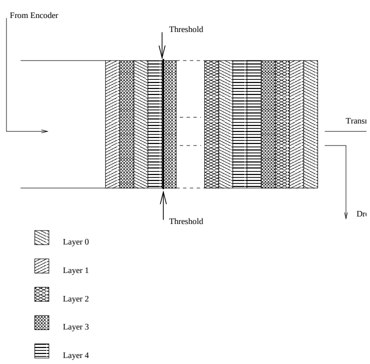
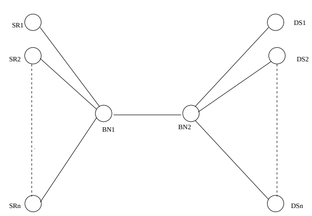


Fig. 1. General Integrated Model



(a) Source Buffer Management



(b) Topology

Fig. 2.

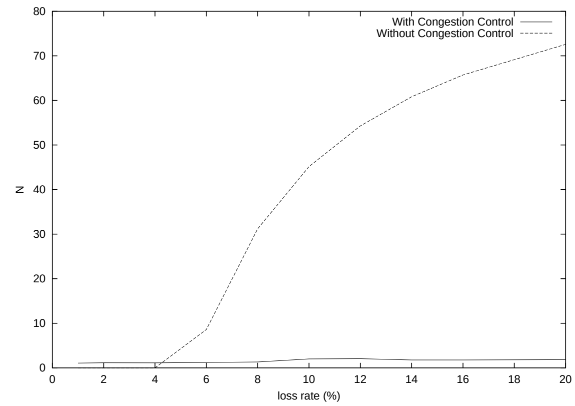


Fig. 3. Average *network loss* for N cbr sources

Algorithm 1 Algorithm for Source Buffer Management

for Each Packet Arrival **do**
  calculate the queue size q_{sz}
  if $q_{sz} > max_q$ **then**
    Drop the packet
  else
    if $q_{sz} > max_{th}$ **then**
      with probability p_k, drop the packet
    else
      enqueue the packet.
    end if
  end if
end for

Algorithm for estimating drop probability p_k
for every RTT **do**
  $lossestimate = (r_e - r_n)/re$
  $lossestimate = lossestimate * MAXLAYER$
  if $lossestimate > 0$ **then**
    for $K = MAXLAYER$; $k > 0$; $k - -$ **do**
      $p_k = lossestimate$
      if $p_k > 1$ **then**
        $p_k = 1$
      end if
      $lossestimate -= p_k$
      if $lossestimate <= 0$ **then**
        break
      end if
    end for
  end if
end for

produces packets with different importance like I, P, B packets of MPEG or layered structure of sub-band wavelet codecs. So video codecs would like the network to treat each packet differently to obtain optimal performance. An integrated design that combines the encoder with the congestion control scheme is therefore required to produce effective video transmission.

Figure 1 shows an integrated model that can achieve the requirements. A video sequence is coded using a robust coding scheme that can adapt to the change in network rate. An adaptive buffer at the sender is used if the encoder cannot adapt as quickly as the network conditions vary. This buffer can be used to differentiate the packets according to their priority and send only the most important packets in the available bandwidth. The buffer will get feedback from the congestion control scheme about the current network conditions. We also need a congestion control scheme that will provide a smooth rate to the video, avoids burst losses and also be fair to other flows in the network. The receiver side has a play-out buffer which will also help in smoothening the flow and reduce jitter. We use randomized binomial schemes for transport as explained in subsequent sections.

B. Avoiding loss of important packets

From extensive simulation results we infer that most of the losses in a TCP based network take place at point of transmission i.e. the source and not at the nodes in the network. This is contrary to the belief that congestion causes the network to drop packets and this dropping of packets in the network is a major contributor to the total loss a TCP flow suffers. Our simulations show that in response to congestion the transmission queues at the sources increase which finally leads to packet drops at source and it is this dropping at the source that is a major contributor to the aggregate loss of the flow. Figure 3(a) shows the average network loss rate with congestion control (TCP) and without congestion control (UDP) with a single bottleneck of 5 Mbps bandwidth for N flows. This clearly provides an incentive to use a suitable congestion control for video transmission because we can control the packets that are dropped at the source buffer by employing a suitable packet filter.

The packet filter can be designed in conjunction with the video codec being used. In this section we propose a simple buffer management strategy that can be used for sub-band/wavelet-decomposed video signal. As shown in figure 2(a) we implement a buffer at the source that queues packets from the encoder and dequeued packets are transmitted using the binomial scheme. Algorithm 1 gives the pseudo code for the drop policy. Whenever the current network rate r_n is less than the application rate r_e, the flow will lose packets at the rate of $r_e - r_n$ packets per second. This drop probability is applied to the packets that arrive at the source buffer such that the packets from the lowest priority layer gets dropped before the next higher priority layer and so on. Each layer is assumed to have equal number of bits. We can also have a running average that calculates how much each layer contributes to the overall bit rate and distribute loss probability accordingly. The packets are dropped only when the buffer reaches a threshold value.

This strategy is of buffer management for the video encoder we are using is supported both by theory and practice.

- **Theory**: S. Mallat [5] shows that coefficient magnitudes of a sub-band/wavelet coded video sequence decrease exponentially for higher layers. So their contribution to PSNR is progressively smaller. Therefore dropping the higher layer first is justified.
- **Practice**: Masry and Hemami [6] found that when bit rate is held constant, lower frame rate is preferred by subjective observers, (i.e) it is better to have higher quality low frame rate video, than lower quality higher frame rate video. By dropping packets belonging to higher layers, we are in effect producing a lower frame rate video with high quality.

Choosing the buffer threshold is very important for performance. Having a small threshold will lead to unnecessary packet drops while having a threshold greater than the receiver buffer (for pre-buffering) will lead to large delay and jitter. Threshold value depends on the decoding rate, network capacity and receiver side buffer. We have chosen some representative values for our simulation. Detailed analysis of these complex interactions is left for future research.

C. Loss of Synchronization and concentration of important information in few packets

In our work we use a robust motion compensated (MC) 3-D sub-band video coder from [1]. The typical group-of-pictures (GOP) structure of this coder is shown in Figure 4. The top level of frames represents the video at full frame rate. Neighboring frames are decomposed using a MC filter bank to produce temporal low frequency bands (solid lines) and temporal high frequency bands (dashed lines) at the next level. Motion vectors are symbolically shown as arrows. High temporal frequency frames are sub-band coded, as described below, and transmitted along with the motion vectors. Low temporal frequency bands at the second level represent the MC average of neighboring frames at the full frame rate, so they occur at $1/2$ of the full frame rate. They are further decomposed to get the video at $1/4$ frame rate, etc. In Figure 4, the last level corresponds to $1/16$ of the full frame rate. Transmitted data in this case is naturally divided into five layers of temporal scalability, labeled (1) through (5) in the figure. Decoders which receive layer (1) can reconstruct the video at $1/16$ of the full frame rate, those which receive (1) and (2) can reconstruct the video at $1/8$ of the full frame rate, and so on.

To enable frame rate scalability, each layer is packetized independently so that the video at lower frame rates can be reconstructed from a subset of the packets corresponding to higher frame rates. Within each layer, data is coded and packetized in a dispersive manner, so that sub-band samples from the common space-time-frequency neighborhood appear in different packets, which enables easy error concealment of lost samples from the available neighboring samples. Also, all the packets from a given layer carry approximately the same amount of information about every frame in that layer, which minimizes the variation of video quality at a fixed packet loss rate. It was found that this dispersive packetization and error concealment approach to robust coding effectively combats the packet loss and eliminates the need for forward error correction (FEC) at packet loss rates up several percent. The scheme can be adapted to change the encoding rate and run-time with some latency.

The simulation studies done for this paper show that in spite of using an advanced robust motion compensated coder, there is a lot of room for improvement that we achieve using our simple scheme of buffer management. Also, FEC fails to solve the problem of performance degradation due to packet drops at the source and as such the coding schemes using FEC do not necessary maximize the performance. The encoding scheme we use is much advanced and has better performance characteristics in the presence of lossy network than the conventional encoding schemes like MPEG [7]. On the basis of simulation results that show that even this robust encoding scheme when coupled with intelligent buffer management and suitable congestion control scheme shows a marked performace increase. We thus propose our buffer management scheme as the general performance improving scheme for any codecs that code in a manner that the resulting packets can be differentiated in discrete levels on the basis of the content or the type of information they carry. An example is the classification of MPEG video traffic into I,B and P frames.

D. Burst Losses, Jitter and TCP-friendliness

In [3] authors propose a family of binomial congestion control schemes that are ideal for multimedia transfer. The advantage of these algorithms is that the reduction in transmis-

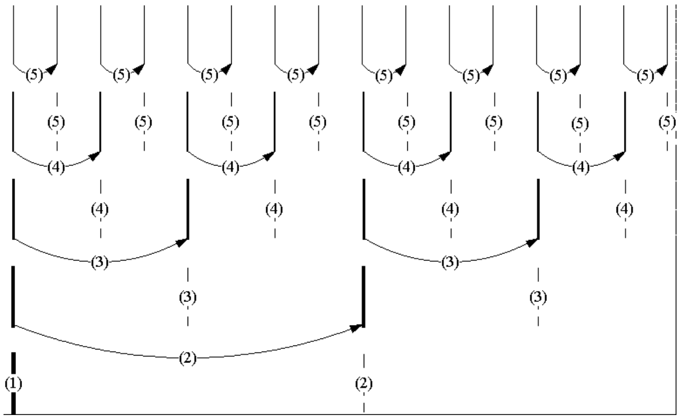


Fig. 4. GOP structure of the video coder

sion rate upon encountering congestion is not as drastic as the conventional TCP. These algorithms use a generalized form of TCP's additive increase policy by increasing the congestion window in steps inversely proportional to a power k of the current window ($k = 0$ for TCP). They also generalize the TCP's multiplicative decrease policy by decreasing proportional to a power l of the current window ($l = 1$ for TCP). The authors [3] show that if $k + l = 1$, the schemes compete fairly with TCP and the class of algorithms is named binomial algorithms. It is further shown that if $k + l > 0$, $k > 0$,$l > 0$ the binomial schemes converge to fairness under a synchronized feedback assumption.

the general increase/decrease equations for binomial algorithms are given as:

- Increase: $w_{t+R} \longleftarrow w_t + \alpha/w_t^k; \alpha > 0$
- Decrease: $w_{t+\delta t} \longleftarrow w_t - \beta w_t^l; 0 < \beta < 1$

where w_t refers to the congestion window size at time t, R is the RTT at t and α and β are constants.

We show by simulation results that the choice of transport scheme matters in multimedia transmission. The performance of the integrated scheme we present is highly dependent on the congestion control algorithm used by the transport scheme. We report our results with the randomized (as explained in the following paragraph) versions of IIAD (Inverse Increase and Additive Decrease) scheme with $k = 1$ and $l = 0$ in the equations above and AIMD (Additive Increase and Multiplicative Decrease) scheme with $k = 0$ and $l = 1$ in the equations above. The simulation results show a clear improvement in performance with IIAD as compared to AIMD.

The randomization concept was first introduced in [4]. Randomization is shown to reduce bias against flows with higher RTTs, reduce window synchronization, reduce phase effects in flows and reduce correlated losses. Randomization does not allow a congestion control scheme send back-to-back packets but spaces successive transmissions with a time interval $\Delta = RTT(1+x)/cwnd$, where x is a zero mean random number drawn from an uniform distribution.

III. Simulation Model

We simulated our integrated model using *ns-2*. The robust encoder was coupled with the transmission protocol we used. The robust codecs discussed in this paper produce a packetized bit stream that is fed to the buffer at the congestion contorol. The encoding scheme employed for the results reported in this paper is such that the coding is done in a hierarchical manner with each frame coded in multiple layers of decreasing content importance. Layers are numbered such that the higher numbered layer are less important than the lower numbered layers. We employ an intelligent transport scheme that differentially drops the higher numbered layer packets in response to reduced offered capacity by the network. The transport scheme we employed uses a randomized version [4] of the binomial schemes proposed in [3]. The randomized binomial schemes provide a less bursty, TCP friendly transport scheme for the media transfer.

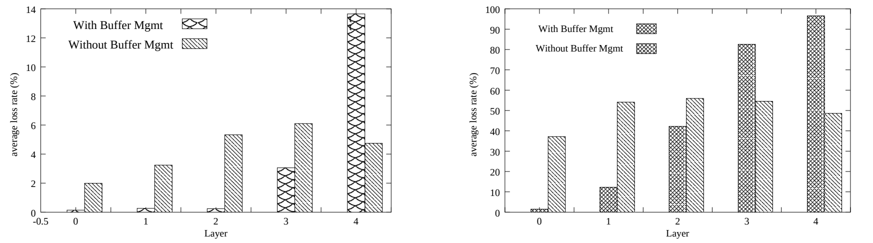
The *ns-2* set-up for all the simulations is the simple dumbbell configuration which is shown in the generalized form with a single bottleneck in figure 2(b). The nodes $SR1$ to SRn are the source stations that transmit the media files using different transport/buffering schemes as dictated by the case configuration. The media flows pass through the bottleneck link BN, finally terminating at nodes $DN1$ to DNn. The access link bandwidths are set at 4 times the bottleneck bandwidth length. The round trip time for all the simulations are set at 100 ms and the buffer at bottleneck isset at one bandwidth-delay product.

Logically each sender is configured as a layered architecture with the movie sequence being fed to the encoder and the encoded file is queued in the transmission buffer where we apply the drop policy as described in algorithm 1. The details of the layered architecture are shown in figure 1.

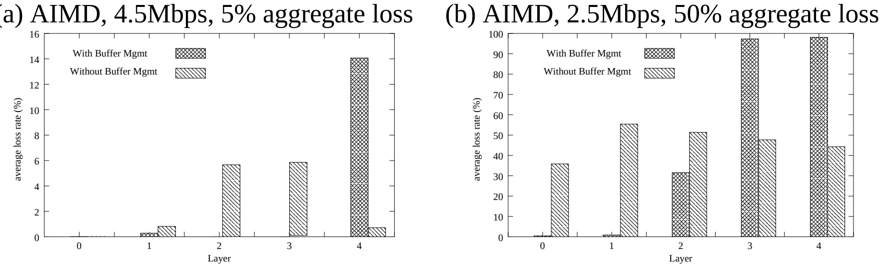
IV. Results

The following results are for 5 source simulation with dumbbell topology shown in figure 2(b). The congestion control scheme, bottleneck bandwidth, maximum transmission buffer and buffer management scheme are the different parameters. The video source produces packets at a rate of 915 kbps. The receiver has a playout buffer of 4 GOPS (800 pkts). Simulations were run to test the performance difference with and without buffer management. All the 5 sources employ the same congestion control scheme and buffer management scheme. The buffer threshold was set to 240 pkts. The figures 5(a), (b), (c) and (d) compares the average loss rate of the 5 flows per layer with and without buffer management. The results show that with buffer management we can intelligently drop more of the low priority layer 4 packets whereas without buffer management, the loss rates are distributed randomly and across all layers. This helps in improving the quality(PSNR) of video. Figure 5(a) is for AIMD scheme with 4.5 Mbps bandwidth. This translates to a aggregate loss rate of about 5 %.5(b) is for AIMD with 2.5Mbps bandwidth. In this case, each source gets only half its required bandwidth. Without buffer management, we will lose approximately 50% of packets in all layers whereas with buffer management, we can drop almost all of layer 3 and 4 packets and get a better PSNR. Figures 5(c) and 5(d) shows the same for IIAD scheme.

The next step of the simulation was to compare the performance of different congestion control schemes. Figures 6(a) and 6(b) compares the average loss rate per layer for AIMD

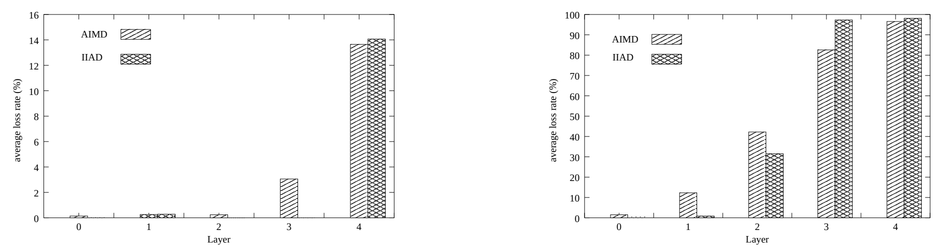


(a) AIMD, 4.5Mbps, 5% aggregate loss (b) AIMD, 2.5Mbps, 50% aggregate loss

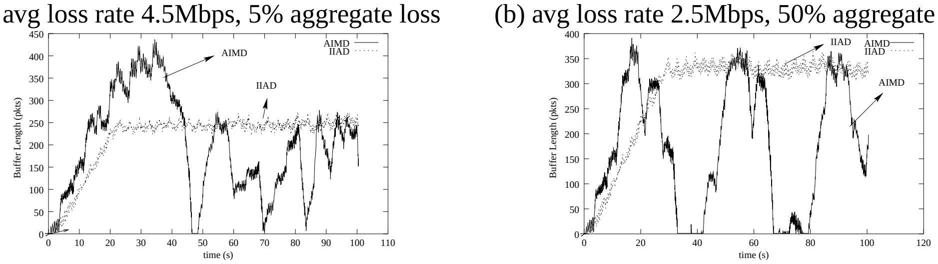


(c) IIAD, 4.5Mbps, 5% aggregate loss (d) IIAD, 2.5Mbps, 50% aggregate loss

Fig. 5. Comparison of average loss rates per layer with and without buffer management



(a) avg loss rate 4.5Mbps, 5% aggregate loss (b) avg loss rate 2.5Mbps, 50% aggregate loss



(c) Buffer Length, 4.5Mbps, 240 pkts threshold (d) Buffer Length, 4.5Mbps, 320 pkts threshold

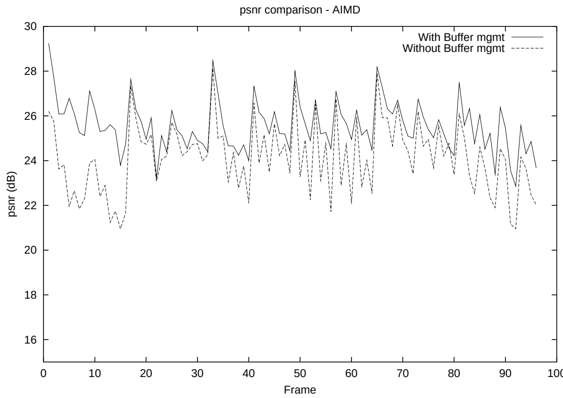
Fig. 6. Comparison of AIMD and IIAD performances

and IIAD schemes. The smoother rate variation of IIAD means more number of layer 4 packets are dropped than higher priority layers. These results show that a smoother rate variation helps the video transmission better. This is because the jitter effects are reduced when the transmission rate do not vary much. The figures 6(c) and 6(d) compares the source buffer length for AIMD and IIAD for different maximum buffer lengths (480 packets and 640 packets respectively). These graphs show that the buffer length for AIMD varies widely resulting in higher jitter at the receiver. Packets can get variable delays. But IIAD maintains the buffer length and reduces any jitter by keeping the delay steady.

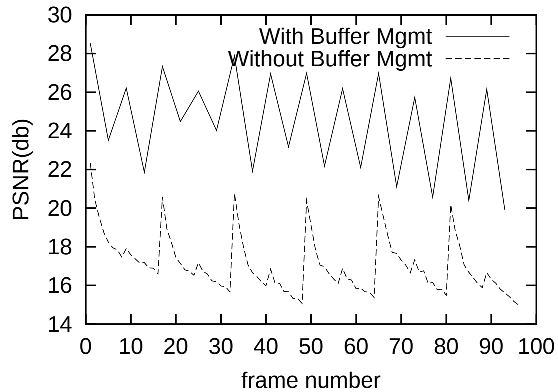
Figure 7(a) compares the PSNR values of the video with and without buffer management. The figure corresponds to simulations with AIMD source, 4.5 Mbps bottleneck bandwidth and 320 pkts source buffer threshold. If more than half the number of packets are lost in any layer, the whole layer is ignored and the resultant video is decoded at lower frame rate. For example if all the packets of layer 5 are dropped, we decode at 1/2 the frame rate. PSNR for the low frame-rate sequence is obtained by comparing the decoded video to the sequence that was passed through motion-compensated temporal filter and coded at a high rate (over 5 Mbps). The graph plots the average PSNR of each frame for 16 seconds of simulation. The average loss rates are comparable (approximately 7%) in both the cases. We can see from the graph that for similar average loss rates, buffer management gives better PSNR than without buffer management. Figure 7 (a) compares PSNR when the loss rate is almost 50%. We can see that in this case, the PSNR values with Buffer Management is much better than that without buffer management. Figures 7 (c) and (d) are snapshots of one frame in IIAD simulation with 4.5Mbps and 240 pkts buffer threshold.We can see that buffer management helps is losing the “correct” packets so as to not affect the quality.

V. Conclusions

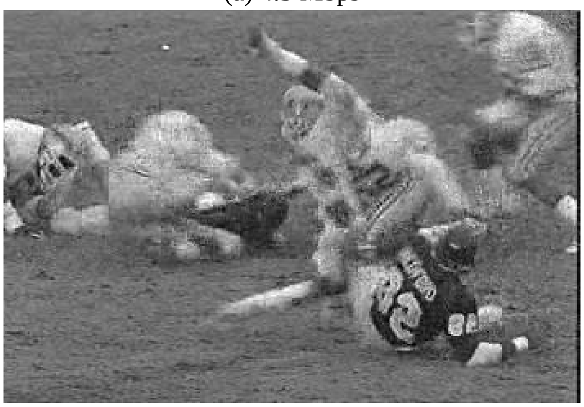
We examined various source-side issues with video streaming over the Internet. An integrated model that controls the



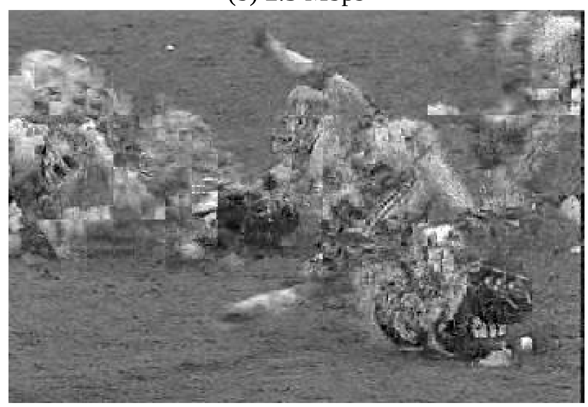
(a) 4.5 Mbps



(b) 2.5 Mbps



(c) With Buffer Management



(d) Without Buffer Management

Fig. 7. PSNR values

packet drops at the source, along with an intelligent choice of congestion control and codec was proposed to solve these problems. It was shown that most of the packet drops occur at the source buffer when using a congestion control scheme. This allows us to design a suitable buffer management scheme for the video flow. A simple packet filter was proposed for the motion compensated wavelet decomposition based encoder. Different congestion control scheme were examined and it was shown that binomial congestion control schemes that do not vary their rates very much along with pacing (randomization) helps reducing jitter effects. It was found that dispersive packetization and error concealment approach to robust coding effectively combats the packet loss and reduces the need for forward error correction (FEC). It was further shown that the performance improvement achieved by using the robust codec mentioned above can be improved by using the integrated model proposed.

REFERENCES

[1] I. V. Bajić and J. W. Woods, ”Domain-based multiple description coding of images and video,” *Proc. SPIE 4671 (VCIP’02)*, pp. 124-135, January 2002.

[2] N. Feamster, D. Bansal, H. Balakrishnan, “On the interactions between layered quality adaptation and congestion control for streaming video,” *11th International Packet Video Workshop (PV2001)*, April 2001.

[3] D. Bansal, H. Balakrishnan, “Binomial Congestion Control Algorithms,” *Proc. IEEE INFOCOM Conf.*, April 2001.

[4] K. Chandrayana, S. Ramakrishnan, B. Sikdar, S. Kalyanaraman, O. Tickoo, A. Balan, “On Randomizing the Sending Times in TCP and other Window Based Algorithms,” *RPI ECSE Networks Laboratory Technical Report, ECSE-NET-2001-1,* July 2001.

[5] Stephane Mallat, “A Wavelet Tour of Signal Processing,” *Academic Press* 1998.

[6] M. Masry, S.S. Hemani, “An Analysis of Subjective Quality in Low Bit Rate Video,” *Proc. ICIP’01* Thesaloniki, Greece, 2001.

[7] I. V. Bajic, J. W. Woods, and A. M. Chaudry, “Robust transmission of packet video through dispersive packetization and error concealment,” *Proc. Packet Video Workshop,* Cagliari, Sardinia, Italy, May 1-2, 2000.

[8] M.Allman and V. Paxson, “TCP Congestion Control,” *Internet Engineering Task Force, RFC2581,* April 1999.

[9] S.Floyd, M. Handley, J. Padhye and J. Widmer, “Equation-Based Congestion Control for Unicast Applications,” *Proc. ACM SIGCOMM.* pp. 43-54, September 2000.

[10] V. Jacobson “Congestion Avoidance and Control,” *Proc. ACM SIGCOMM*, pp. 314-329, August 1988.

[11] K.-W. Lee, R. Puri, T. Kim, K. Ramchandran and V. Bharghavan, “An Integrated Source Coding and Congestion Control Framework for Video Streaming in the Internet,” *IEEE Infocom 2000,* March 2000.

[12] R. Rejaie, M. Handley and D. Estrin, “Layered Quality Adaptation for Internet Video Streaming,” *IEEE Journal on Selected Areas of Communication (JSAC),* Winter 2000.

[13] I. V. Bajic and J. W. Woods “Domain-based Multiple Description Coding of Images and Video,” *Proc SPIE 4671 (VCIP02)*, pp. 124-135, January 2002.

[14] M. Vishwanath and P. Chou “An efficient algorithm for hierarchical compression of video,” *Proc. IEEE International Conference on Image Processing* November 1994.