Proceedings of the 10th WSEAS International Conference on COMMUNICATIONS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp187-192)

A GoP Based FEC Technique for Packet Based Video Streaming

YUFEI YUAN[1], BRUCE COCKBURN[1], THOMAS SIKORA[2], and MRINAL MANDAL[1,2]
[1]Dept of Electrical and Computer Engg, University of Alberta, Edmonton, CANADA
[2]Institut für Telekommunikations systeme, Technische Universität Berlin, Berlin, GERMANY
yfyuan@gmail.com, cockburn@ece.ualberta.ca, sikora@nue.tu-berlin.de, mandal@ece.ualberta.ca

Abstract: - In this paper, we propose an efficient forward error correction (FEC) technique for video transmission in a lossy network. Here, the FEC is applied on source packets at group of pictures level assuming an MPEG-like compression scheme. We also derive analytically an estimate of the playable frame rate for the proposed technique. It is shown, by both analysis and simulation, that the proposed FEC technique provides a better playable frame rate than the classical frame-level FEC techniques.

Key-Words:- Video Streaming, Forward Error Correction, MPEG, Group of Pictures, Playable Frame Rate

1 Introduction

Video traffic through the Internet has increased significantly in recent years. Typical video applications include news broadcasts, video clips, music television, and video conferencing. However, the Internet has limited bandwidth, and excessive traffic may lead to congestion at times. Videos are generally transmitted through Internet using packets. When congestion occurs in the network, some video packets are likely to be lost.

Several techniques have been suggested to solve the packet loss problem. The forward error correction (FEC) scheme [1] has been shown to be an effective way to combat packet loss during video streaming. In FEC, redundant packets are transmitted along with source packets. If the number of lost packets is smaller than the number of redundant packets, the video data can be reconstructed without error. In this paper, we consider developing an efficient FEC-based transmission mechanism.

Several FEC-based techniques have been proposed in the literature. Mayer-Patel *et al.* [2] presented an analytical FEC model for the MPEG frame structure that uses three types of frames (I, P, and B). Wu *et al.* [3] extended this model and derived analytically the playable frame rate (PFR) for a given packet loss probability. However, these techniques assume that the FEC coding rate is allocated among the different picture types. This allocation strategy is not necessarily the best strategy for packet-based FEC in MPEG framework.

In this paper, we propose a FEC technique for video streaming. Here, the FEC is applied at the group of pictures (GoP) level instead of being applied only at the frame level. The average playable frame rate (PFR) for the proposed FEC technique is derived analytically. It is shown, by both analysis and simulation, that the proposed FEC technique provides a better PFR than the frame-level FEC technique.

The paper is organized as follows. Section 2 presents a brief overview of the current FEC techniques. The proposed analytical model is then derived in Section 3. Performance of the proposed technique is evaluated in section 4, which is followed by the conclusions in Section 5.

2 Review of Background Work

In this section, we present a brief review of the background work on FEC based video streaming.

When video packets are sent through a lossy channel, some packets are likely to be lost. This packet loss is generally modeled as Bernoulli trials. When K source packets are transmitted with (N-K) redundant packets, the probability of successful transmission is given by [4]

$$B(N,K,p)=\sum_{r=K}^{N}\binom{N}{r}(1-p)^r p^{N-r} \tag{1}$$

where p is the packet loss probability.

Current video coding standards such as MPEG uses the so called hybrid coding where redundancy in the frames is first removed by motion compensation. Further redundancy reduction is then obtained using block based discrete cosine transform. Note that in MPEG video coding, there are three types of frames (I, P, and B) as shown in Fig. 1.

Wu *et al.* [3] have recently proposed an analytical model (henceforth referred to as the frame-level FEC technique) for optimizing FEC-based transmission in the GoP based MPEG framework. The FEC packets are generated based on individual frames (I, P, or B) as shown in Fig. 2.

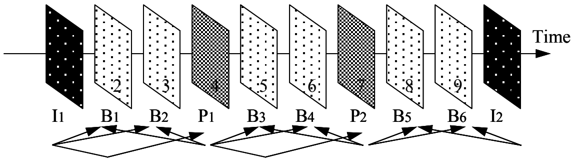


Fig. 1: The structure of a GoP and the inter-frame dependency relationship within it.

If each GoP includes one I-frame, N_P P frames and N_B B frames, the effective GoP transmission rate is given by

$$G = \frac{T / s_{pkt}}{(S_I + S_{IF}) + N_P(S_P + S_{PF}) + N_B(S_B + S_{BF})} \qquad (2)$$

where T is the transmission rate, s_{pkt} is the packet size, S_I is size of I frames (in packets), S_P is size of P frames (in packets), S_B is size of B frames (in packets), S_{IF} is the number of FEC packets added to each I frame, S_{PF} is the number of FEC packets added to each P frame, S_{BF} is the number of FEC packets added to each B frame.

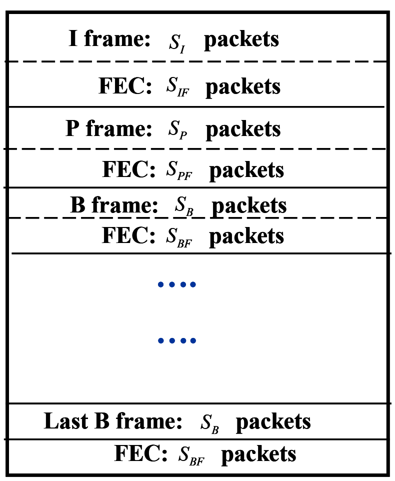


Fig. 2: Arrangement of source and FEC Packets in frame-level FEC technique.

The total PFR by Wu *et al.*'s technique is given by

$$R = G.q_I \left[1 + \frac{q_P - q_P^{N_P+1}}{1 - q_P} + N_{BP}.q_B \left(\frac{q_P - q_P^{N_P+1}}{1 - q_P} + q_I q_P^{N_P} \right) \right] \qquad (3)$$

where q_I, q_P, and q_B are the probabilities of successful transmission of an I, P, or B frame, respectively. The probabilities q_I, q_P, and q_B can be expressed as follows:

$$\begin{aligned} q_I &= B(S_I + S_{IF}, S_I, p) \\ q_P &= B(S_P + S_{PF}, S_P, p) \\ q_B &= B(S_B + S_{BF}, S_B, p) \end{aligned} \qquad (4)$$

where $B(.,.,.)$ is calculated using Eq. (1).

3 Proposed FEC Technique

The frame-based FEC technique provides a good error resiliency performance with appropriate selection of parameters such as S_{IF}, S_{PF} and S_{BF} [3]. A problem with this approach is that the allocation of FEC packets for each type of frames is static. In this paper, we propose a FEC model where we first select, depending on the network condition and the GoP structure, an appropriate number of FEC packets for a GoP. The FEC packets are then generated for the entire GoP and added to the original source packets. The number of redundant packets is added such that the playable frame rate is maximized.

3.1 The Proposed Model

The organization of frames in a typical GoP looks like the following:

$$IB_{0,0}....B_{0,N_{BP}-1}P_1........P_m B_{m,0}....B_{m,N_{BP}-1}P_{m+1}......P_{N_P} B_{N_P,0}....B_{N_P,N_{BP}-1}$$

Note that N_P is the number of P frames, and N_{BP} is the number of B frames between two successive reference frames. The number of B-frames in a GoP is given by $N_B = (1 + N_P)N_{BP}$. Therefore, the total number of source packets is given by

$$K = S_I + N_P \cdot S_P + (N_P + 1) \cdot N_{BP} \cdot S_B$$

where S_I, S_P, and S_B are the size of I, P and B frames (in packets), respectively. These K source packets are arranged as shown in Fig. 3. Note that the frames have been arranged as per their predictive behavior.

We now add (*N*-*K*) FEC packets, resulting in a total of *N* (source and FEC) packets. We propose to use a class of linear erasure codes [5] known as systematic codes. For systematic (n, k) codes, the $k \times n$ generator matrix includes the identity matrix $(k \times k)$ as a sub-matrix. As a result, the FEC coded data packets include the source data packets. This will provide two advantages. When the number of lost packets is less than or equal to (*N*-*K*), the entire GoP can be recovered. Even when the number of lost packets is greater than (*N*-*K*), the GoP can be partially recovered. The advantage of this model over frame-based model is explained by an example below.

Assume that a GoP has 72 source packets, which includes 24 source packets from the I frame. Further assume that the number of FEC packets is 20, and in the frame-based technique 6 out of these 20 packets correspond to the I-frame. The total number of packets (source + FEC) is therefore 92. During the transmission, let us assume that 15 packets are lost. The GoP based technique can easily reconstruct the entire GoP. However, the performance of the frame-based technique will depend on the frames related to the lost packet. If 10 out of these 15 lost packets belong to the I frame, the I frame cannot be reconstructed, and the entire GoP is virtually lost.

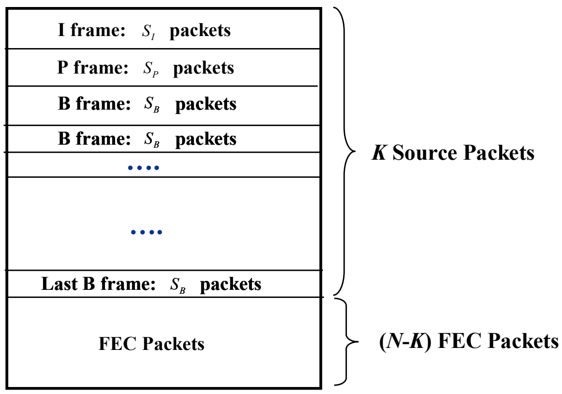


Fig. 3: Arrangement of source and FEC Packets in the proposed technique.

It is possible to come up with a counter example where the frame-based technique will perform better than the GoP based technique. Therefore, we derive an analytical formula, in section 3.2, for the playable frame rate (PFR) of the GoP based model. This can then be compared with the PFR of the frame-based model to determine the effectiveness of the proposed model.

3.2 PFR for GoP-based FEC

In order to calculate the overall PFR, we calculate the decoding probabilities of I, P and B frames, which are denoted by q_I, q_P, and q_B, respectively. The calculation of these probabilities is explained below.

In order to calculate q_I (the probability of successfully delivering an I-frame), we classify the delivery of N packets into three situations based on the number of lost packets L.

a) The I-frame is decodable when $L \le N-K$.
b) The I-frame is decodable with certain probability when $N-K < L \le N-S_I$.
c) The I-frame is not decodable when $L > N-S_I$.

Note that in case (a), L is smaller than or equal to the number of redundant packets. Therefore, this case is fully protected with (N,K) systematic codes, and we should not experience any decoding error. In case (b) L exceeds the number of redundant packets, and we will have decoding errors. However, if L is smaller than $(N-S_I)$, there is a possibility that all lost packets belong to packets from P and B frames (or redundant packets). In case (c), too many packets have been lost, and therefore I frame is not decodable.

Combining all three situations, the probability that I-frame is playable, can be expressed as

$$q_I = \sum_{r=0}^{N-K} \binom{N}{r} p^r (1-p)^{N-r} + (1-p)^{s_I} \sum_{r=N-K+1}^{N-s_I} \binom{N-s_I}{r} p^r (1-p)^{N-s_I-r}$$
$$= B(N,K,p) + (1-p)^{s_I} B(N-s_I, N-K+1, 1-p)$$

The playable rate of I frames can be expressed as

$$R_I = G.q_I \tag{5}$$

Now consider a P frame. The *m*th P frame, P_m is playable if it's preceding I and P frames, and itself are successfully transmitted. Assuming that the previous reference frames are available, we have the following three situations with respect to the number of lost packets L.

a) The P_m-frame is decodable when $L \le N-K$.
b) The P_m-frame is probably decodable with certain probability when $N-K < L \le N-S_I-mS_P$.
c) The P_m-frame is not decodable when $L > N-S_I-mS_P$.

Therefore, the play rate of the *m*th P-frame (P_m) is given by

$$q_{P_m} = \sum_{r=0}^{N-K} \binom{N}{r} p^r (1-p)^{N-r}$$
$$+ (1-p)^{s_I+ms_P} \sum_{r=N-K+1}^{N-s_I-ms_P} \binom{N-s_I-ms_P}{r} p^r (1-p)^{N-s_I-ms_P-r}$$
$$= B(N,K,p) + (1-p)^{s_I+ms_P} B(N-s_I-ms_P, N-K+1, 1-p)$$

The playable rate of P frames can be expressed as

$$R_P = G.q_P = G.\sum_{m=1}^{N_P} q_{P_m} \tag{6}$$

For all B frames except those after the last P-frame, we have the following three situations with respect to the number of lost packets L.

a) The $B_{i,j}$-frame is decodable when $L \le N-K$.
b) The $B_{i,j}$-frame is decodable with certain probability when $N-K < L \le N-s_I-(i+1)s_P-s_B$.
c) The $B_{i,j}$-frame is not decodable when $L > N-s_I-(i+1)s_P-s_B$.

The probability of successful decoding of these B-frames is given by [6]

$$q_{B_{i,j}} = (1-p)^{s_I+(i+1)s_P+s_B} B(N-s_I-(i+1)s_P-s_B, N-K+1, 1-p) + B(N,K,p) \quad (7)$$

For the B frames after the last P-frame (i.e., those preceding the I-frame of the next GoP), the successful decoding is possible only if both the P_{N_P}-frame (in the current GoP) and the I-frame of the next GoP are successfully decoded. Therefore, the probability of successful decoding of these B-frames is given by [6]

$$q_{B_{N_P,j}} = q_I \begin{bmatrix} (1-p)^{s_I+N_P s_P+s_B} B(N-s_I-N_P s_P-s_B, N-K+1, 1-p) \\ +B(N,K,p) \end{bmatrix} \quad (8)$$

Combining (7) and (8), the playable rate of B frames can be expressed as

$$R_B = G.q_B = G.N_{BP} \sum_{i=0}^{N_P} q_{B_{i,0}} \quad (9)$$

The total playable frame rate is expressed by

$$R = R_I + R_P + R_B = G\left(q_I + q_P + q_B\right) \quad (10)$$

where R_I, R_P, and R_B are calculated using Eqs. (5), (6) and (9), respectively.

The complexity of the packet generation depends on the FEC codes used. In this paper, we assume that the erasure codes are systematic codes, and hence only the redundant packets need to be generated. It can be shown that [17] for a typical compression framework, the complexity of the proposed technique is about 5-7 times that of Wu's technique [3]. However, since the generation of redundant packets makes up only a small part in the computation of the streaming applications, we do not expect a significant impact on the overall computational complexity of the codec by replacing a frame-based FEC technique with a GoP-based FEC technique.

4 Performance Evaluation

In this section, the closed form formula derived in section 3 is compared with that of the frame level FEC model in [3]. The PFR computed from using the two models will be compared in section 4.1. In section 4.2, we will use a non-scalable MPEG-4 trace and the NS-2 network simulator [7] to conduct FEC simulations for video streaming.

4.1 Model-based Analysis

We calculate the PFR of the proposed technique and compare it that of [3]. The PFR in Eqs. (3) and (10) provides the rate in frames/sec. For simplicity, we can also express the PFR as a ratio (in %) as follows.

$$\text{PFR Ratio (in \%)} = \frac{\text{PFR}}{\text{Source Frame Rate}} \times 100 \quad (11)$$

The PFR in Eq. (11) provides the percentage of the frames in a GoP that can be decoded correctly at the receiver. The source frame rate typically varies between 15 and 30 depending on the applications. In the simulation, we assume a rate of 25 frames/sec to calculate the PFR ratio.

The network settings, such as packet size (s_{pkt}), the round-trip time (t_{RTT}), TCP retransmit timeout value (t_{RTO}) are taken from typical network connections. We assume the UDP as the transport protocol. However, in order to avoid network congestion, we assume that the UDP transmission is TCP-friendly. We use the following formula to calculate the transmission rate in the network [4] for a given packet loss probability.

$$T = \frac{s_{pkt}}{t_{RTT}\sqrt{\dfrac{2p}{3}} + t_{RTO}\sqrt{\dfrac{27p}{8}}\, p\left(1+32p^2\right)} \quad (12)$$

The bitrate of a streamed video is highly variable, and can range from 64 Kbps to 10 Mbps. In this analysis, the bitrate is set at 1.15 Mbps (MPEG-1 VCD quality). A GoP is assumed to have 12 frames with $N_P = 3$, and $N_{BP} = 2$. The parameter values used in this analysis are t_{RTT}=50 ms, t_{RTO}=200 ms, s_{pkt}=500 or 1000 bytes, and the network loss probability, p = [0.005,0.01,0.02,0.03, 0.04,0.05,0.06,0.07, 0.08,0.09, 0.10]. Note that a bitrate of 1.15 Mbps will result in approximately 300 and 150 packets/sec for packet size 500 and 1000 bytes, respectively.

Fig. 4 shows the PFR of the frame-based as well as GoP based techniques with no FEC. The plot was generated using Eqs. (3), (10) and (11). Since there is no FEC in the streaming, the performance of the frame-based FEC and the GoP-based FEC is identical. It is observed that the PFR deteriorates very quickly. We can see that only around 40% of the frames can be delivered error free at p=0.02, which is a fairly small packet loss probability. Fig. 4 also shows the effect of packet size on the PFR. In order to keep a constant channel-coding rate, the number of packets was doubled when s_{pkt}=500. It is observed that the streaming performance with s_{pkt}=1000 is significantly better. This is mainly because, with the same packet loss probability, the playable frame rate is statistically better (see Eq. (3)) when the number of packets is small. When s_{pkt}=500, we use more packets, and therefore, the PFR drops.

Fig. 5 shows the improvement of PFR when FEC is added. The symbol (S_{IF}, S_{PF} and S_{BF}) corresponds to the number of FEC packets for I-, P-, and B-frames for

Proceedings of the 10th WSEAS International Conference on COMMUNICATIONS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp187-192)

the frame-based FEC. In other words, the total number of redundant packets (for one GoP) is given by

$$S_{IF} + N_P \cdot S_{PF} + \left(N_P + 1\right) \cdot N_{BP} \cdot S_{BF}$$

In order to compare the frame and GoP based techniques, we add identical number of redundant packets to a GoP source packets.

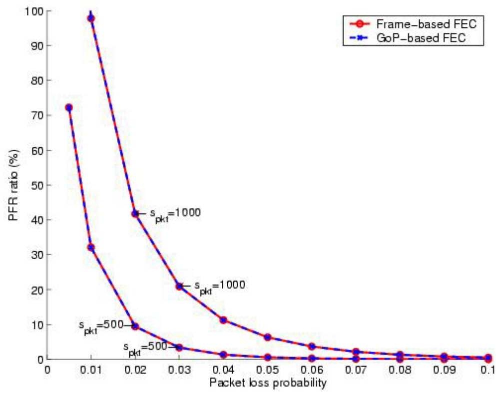


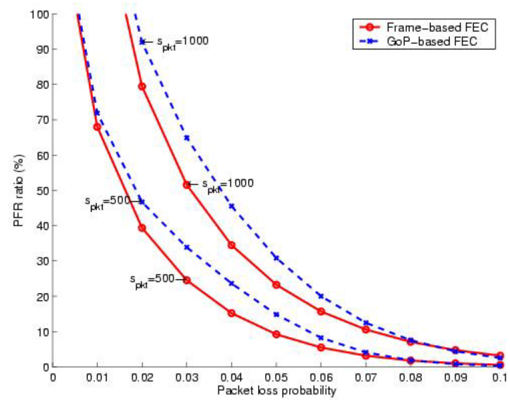
Fig. 4: Comparison of PFR ratio with no FEC for parameters t_{RTT} =50 ms, and t_{RTO} =200 ms.

Fig. 5(a) shows the PFR after we provide a light weight FEC (1,1,0) whereas Fig. 5(b) shows the PFR when moderate are imposed. It is observed that the proposed technique provides a significant PFR improvement over the frame-based FEC technique. It has been found that the proposed technique provides a performance similar to frame-based technique at high FEC, and has not been shown in the figure.

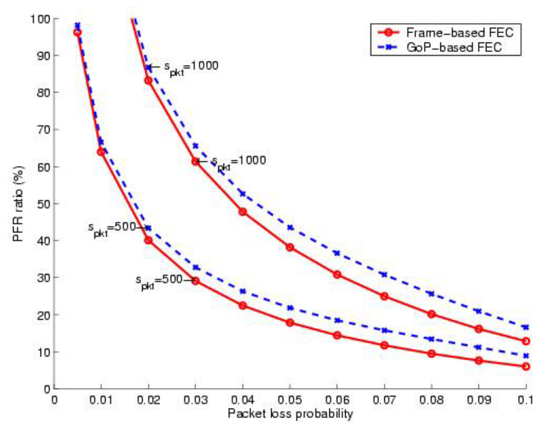
Although, the FEC in general improves the PFR in a lossy network, a heavy weight FEC need not necessarily perform better than a light weight FEC. A close look at Fig. 5(a) and 5(b) will reveal that at p=0.02, FEC (1,1,0) provides a better performance than FEC (4,2,0). This is primarily because of the FEC overhead. If the FEC provided exceeds an appropriate level, it occupies unnecessary extra bandwidth that could have been used to transmit source packets. It has been shown by experiments that for marginally lossy network, a light weight FEC provides the best performance whereas for moderate lossy network, a medium weight FEC provides the best performance. Finally, for the highly lossy network, a heavy weight FEC provides the best performance. In all three situations, the proposed GoP based FEC provides a superior performance compared to the frame-based FEC technique.

4.2 Simulation-based Analysis

In section 4.1, we have compared the performance of the GoP-based and frame-based techniques. The comparison was done analytically with fixed model parameters. However, in practice, network conditions and frame sizes vary statistically at various temporal scales. To obtain a more realistic performance comparison, we have evaluated the performance on NS-2 network simulator. Instead of using a fixed mean compressed frame size to compute the PFR, we used the real downloadable trace files of videos generated by an MPEG-4 encoder [6]. The streaming server reads entries from a trace file, generates source packets, and passes the source packets to the UDP agent for transmission. Since the trace file represents a variable bitrate compressed video bitstream, the client contains a receiver buffer to smooth out the bitstream variations. The decoder then periodically accesses the receiver buffer to retrieve packets for decoding. If all packets for a frame and its reference frame(s) are received, the frame is labeled playable; otherwise, the frame is declared unplayable.



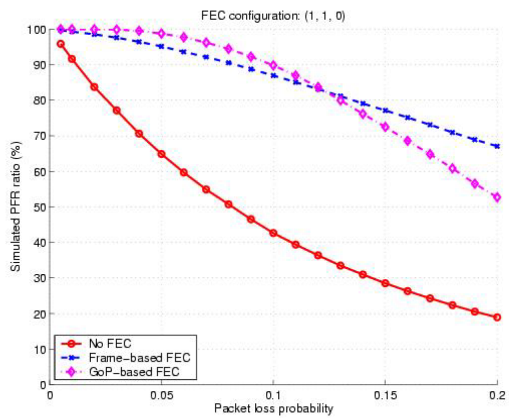
(a) FEC: (1,1,0)



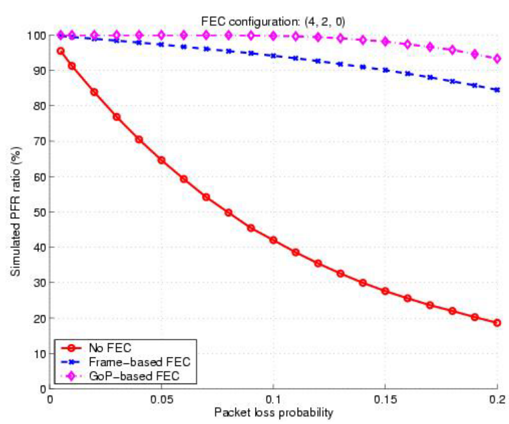
(b) FEC: (4,2,0)

Fig. 5: Comparison of PFR ratio with parameters (t_{RTT} =50 ms, t_{RTO} =200 ms). Note that the number of FEC packets in (a), (b) are respectively, 4, and 10.

In our simulation, a 10 minutes clip was streamed out of the movie "Die Hard III". The movie clip was encoded at medium quality using an MPEG-4 encoder. Independent packet loss events during a streaming session were assumed throughout our simulations. The simulation results with different FEC configurations are illustrated in Fig. 6. For every value of p, we used ten different seed values for the random number generator to generate different loss patterns. In Fig. 6, the mean PFR values are plotted for each p. To show the effectiveness of the FEC, the PFR values without FEC are also plotted for comparison.



(a)



(b)

Fig. 6: The effect of adjusting FEC configuration on the performance of video streaming using a non-scalable MPEG-4 source trace.

It is observed in Fig. 6 that the GoP based FEC technique performs better than the frame-based technique in most cases. It may be apparent from Fig. 6(a) that the frame-based technique provides a better performance than the GoP based technique for FEC (1,1,0) when p exceeds 0.12. However, the FEC configuration (1,1,0) is only optimal near a packet loss probability of 0.005. When the packet loss probability exceeds 0.12, stronger FEC protection such as (4,2,0) should be employed. In that case, the GoP based FEC technique will perform better. In other words, it can be concluded that the GoP based FEC technique always performs better than the frame-based technique when an optimal FEC configuration is used.

5 Conclusions

In this paper a new analytical model is derived to evaluate a media-dependent FEC scheme for video streaming applications. It is shown in the analytical results that in most typical network conditions, the usage of a GoP-level FEC scheme should be preferred over a frame-level FEC scheme. The analytical results are validated by experimental simulations on the NS-2 network simulator. Our model can be used to compute the optimal allocation of FEC for compressed video streams of different rates at a given estimate of the network loss probability. It is clear that the results hold for any type of video data those are compressed by hybrid video encoders.

References:

[1] K. Park and W. Wang, QoS-sensitive Transport of Real-time MPEG Video using Adaptive Forward Error Correction, *Proc. of IEEE Multimedia Systems*, June 1999, pp. 426-432.

[2] K. Mayer-Patel, L. Le, and G. Carle, An MPEG Performance Model and its Application to Adaptive Forward Error Correction, *Proc. of ACM Multimedia*, December 2002, pp. 1-10.

[3] H. Wu, M. Claypool, and R. Kinicki, A Model for MPEG with Forward Error Correction and TCP-friendly Bandwidth, *Proc. of the ACM NOSSDAV*, 2003, pp. 122-130.

[4] J. Padhye, V. Firoiu, *et al*., Modeling TCP Reno Performance: A Simple Model and its Empirical Validation, *IEEE/ACM Transactions on Networking*, Vol. 8, Issue 2, April 2000, pp. 133 – 145.

[5] L. Rizzo, Effective Erasure Codes for Reliable Computer Communication Protocols, *ACM SIGCOMM Computer Comm. Rev*., Vol. 27, No. 2, April 1999, pp. 24-36.

[6] Y. Yuan, *Wavelet Video Coding with Application in Network Streaming*, Ph.D. Thesis, Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada, Spring 2005.

[7] VINT Project, "The Network Simulator – NS-2," http://www.isi.edu/nsnam/ns/.