Journal of Mobile Multimedia, Vol. 3, No.1 (2007) 047-064

EVALUATION OF STREAMING MPEG VIDEO OVER WIRELESS CHANNELS

CHIH-HENG KE, CHENG-HAN LIN, CE-KUEN SHIEH

Department of Electrical Engineering, National Cheng Kung University,
smallko@ee.ncku.edu.tw; jhlin5@hpds.ee.ncku.edu.tw; shieh@ee.ncku.edu.tw

WEN-SHYANG HWANG

Department of Electrical Engineering, National Kaohsiung University of Applied Sciences
wshwang@mail.ee.kuas.edu.tw

ARTUR ZIVIANI

National Laboratory for Scientific Computing (LNCC), Brazil
ziviani@lncc.br

Received January 27, 2006
Revised August 23, 2006

We propose an analytical model to derive from the packet loss rate an objective application-level metric, the decodable frame rate (Q), thus allowing the evaluation of the packet loss effect on the streaming MPEG video quality over wireless channels, as perceived by end users. We analyze the effects on video quality using a random uniform error model and the Gilbert-Elliot (GE) error model (for burst errors modeling) to represent wireless lossy channels. Results obtained through extensive simulations indicate the effectiveness of our proposed model for both the random uniform and GE error models, provided that the packet error rate remains relatively low. Moreover, owing to the well-acceptance of PSNR as an objective performance metric that takes into account the video content to assess the video quality, we also investigate the relationship between PSNR and Q as well as their comparative performances as metrics for video quality assessment. Results show that the Q metric reflects well the behavior of the PSNR metric, whereas being much less time-consuming.

Key words: MPEG, video quality evaluation, wireless
Communicated by: J.-H. Ma & V. Tsaoussidis

1 Introduction

The ever-growing number of users using Internet multimedia services is attracting several Internet researchers to plunge into the studies on QoS for video transmission over lossy channels [1]. Moreover, due to the convenience offered by wireless networks, more and more Internet users choose to connect to the Internet using mobile devices, such as notebooks and PDAs. Nevertheless, the wireless medium offers a much more hostile environment for transmission, thus being characterized by higher bit error rates. Previous studies on the delivered quality to MPEG video streams over wireless error channels generally present the results using only network-level parameters, such as packet/frame delay, packet/frame jitter, and throughput [2]. However, these network-level parameters may be insufficient to accurately rate the video quality perceived by an end user at the application level. Therefore, it is important to map network-level parameters (e.g. packet loss rate) onto meaningful application-level metrics to better reflect video quality as perceived by an end user and capture this quality rating with an objective metric.

In this paper, we propose an evaluation metric, the decodable frame rate (Q), that builds upon outcomes from a previous study [3] to represent the quality result of streaming MPEG video over lossy channels. The main difference between our model and the earlier one is that we use the packet level as an analytical parameter while they use the frame level. The Q metric is an application-level parameter and is thereby more suitable to evaluate the perceived video quality by end users.

Furthermore, owing to the well-acceptance of PSNR as an objective performance metric that takes into account the video content to assess the video quality, we investigate the relationship between PSNR and Q as well as their comparative performances as metrics for video quality assessment. Our simulation-based analysis takes into account the video contents and was enabled by a previous work of ours, the enhanced Evalvid system [4, 5]. This system is being successfully adopted as an evaluation tool for video networked transmission in different recent related work [6, 7, 8]. Results show that the Q metric reflects well the behavior of the PSNR metric, whereas being much less time-consuming—the Q and PSNR metrics run in O(N) and O(kN) time, respectively. Note that N is the number of frames in the video stream, k is the number of pixels within each frame, and the latter is typically of the same order of magnitude as the former.

The remainder of this paper is organized as follows. In section 2, we introduce the analytical model of MPEG video delivery over wireless error channels. Section 3 discusses our experimental settings. Section 4 presents simulation results. Section 5 studies the relationship between PSNR and Q. Finally, we conclude and present future work in Section 6.

2 Analytical Model

Standard MPEG encoders basically generate three distinct types of frames, namely I, P, and B frames [9]. Due to the hierarchical structure of MPEG, I frames are more important than P frames, and in turn P frames are more important than B frames. Therefore, a frame is considered decodable if, and only if, all the fragmented packets of this frame and the other packets that this frame depends on are completely and timely received. Thus, the decodable frame rate (Q) is defined as the number of decodable frames at the receiver over the total number of frames originally sent by a video source, i.e.

$$Q = \frac{N_{dec}}{(N_{total-I} + N_{total-P} + N_{total-B})},$$

where N_{dec} is the summation of N_{dec-I}, N_{dec-P}, and N_{dec-B}, defined in the following. The larger the Q value, the better the video quality perceived by the end user.

- *The expected number of decodable I frames (N_{dec-I})*

In a GOP, the I frame is decodable only if all the packets that belong to the I frame are received and are not too late to be played out. Therefore, the probability of an I frame being decodable is

$$S(I) = \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I},$$

where p_c is the probability for congestion loss, p_w is the probability for wireless error, p_d is the probability for a packet that is too late to be played-out, and C_I is the mean number of packets to transport the data of I frame. Congestion loss, wireless error, and delay/jitter are three main factors to impair the perceived video quality. Thus, the expected number of correctly decodable I frames for the whole video stream is

$$N_{dec-I} = \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I} \times N_{GOP},$$

where NGOP is the total number of GOPs in the video sequence.

- *The expected number of decodable P frames (N_{dec-P})*

In a GOP, the P frame is decodable only if the preceding I or P frames are decodable and all the packets that belong to the current P frame are received as well. And all fragmented P frame packets can not be too late to be played out. Therefore, the probabilities that each of the N_P frames of type P in a GOP is decodable are

$$S(P_1) = \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I} \times \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_P} = \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I+C_P}$$

$$S(P_2) = \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I} \times \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_P} \times \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_P}$$
$$= \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I+2\times C_P}$$

…………..………………………………………………………………………………

$$S\left(P_{N_P}\right) = \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I} \times \left[(1-p_c)(1-p_w)(1-p_d)\right]^{N_P\times C_P}$$
$$= \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I+N_P\times C_P},$$

where C_P is the mean number of packets to transport the data of P frame. Thus, the expected number of correctly decodable P frames for the whole video is

$$N_{dec-P} = \left[(1-p_c)(1-p_w)(1-p_d)\right]^{C_I} \times \sum_{j=1}^{N_P} \left[(1-p_c)(1-p_w)(1-p_d)\right]^{j\times C_P} \times N_{GOP}$$

- *The expected number of decodable B frames (N_{dec-B})*

In a GOP, the B frame is decodable only if the preceding and succeeding I or P frame are both decodable and all the packets that belong to the current B frame are all received. Also, all fragmented B frame packets can not be too late to be played out. As consecutive B frames have the same dependency throughout the GOP structure, we consider consecutive B frames as composing a B group. For the I-to-I frame distance (N), and the I-to-P frame distance (M) GOP structure, each GOP has (M-1) B frames in each B group. We use B_n to represent each of the B frames that compose the n^{th} B group. Therefore, the probabilities of each B frame in the ($\frac{N}{M}-1$) initial B groups to be decodable are

$$S(B_1) = [(1-p_c)(1-p_w)(1-p_d)]^{C_I} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_P} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_B}$$
$$S(B_2) = [(1-p_c)(1-p_w)(1-p_d)]^{C_I} \times [(1-p_c)(1-p_w)(1-p_d)]^{2\times C_P} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_B}$$

……………………………………………………………………………………………..

$$S\left(B_{\frac{N}{M}-1}\right) = [(1-p_c)(1-p_w)(1-p_d)]^{C_I} \times [(1-p_c)(1-p_w)(1-p_d)]^{\left(\frac{N}{M}-1\right)\times C_P} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_B},$$

where C_B is the mean number of packets to transport the data of B frame. The last B group is also affected by the I frame of the succeeding GOP. Hence,

$$S\left(B_{\frac{N}{M}}\right) = [(1-p_c)(1-p_w)(1-p_d)]^{2\times C_I} \times [(1-p_c)(1-p_w)(1-p_d)]^{\left(\frac{N}{M}-1\right)\times C_P} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_B}$$

Given that each B group has (M-1) B frames and $N_P = \frac{N}{M} - 1$, the expected number of correctly decodable B frames for the whole video is

$$N_{dec-B} = (M-1) \times \sum_{j=1}^{\frac{N}{M}} S(B_j) \times N_{GOP}$$
$$= \{ (M-1) \times [(1-p_c)(1-p_w)(1-p_d)]^{C_I} \times \sum_{j=1}^{N_P} [(1-p_c)(1-p_w)(1-p_d)]^{j\times C_P} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_B}$$
$$+ (M-1) \times [(1-p_c)(1-p_w)(1-p_d)]^{2\times C_I} \times [(1-p_c)(1-p_w)(1-p_d)]^{N_P \times C_P} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_B} \} \times N_{GOP}$$
$$= (M-1) \times [(1-p_c)(1-p_w)(1-p_d)]^{C_I} \times [(1-p_c)(1-p_w)(1-p_d)]^{C_B} \times \{ \sum_{j=1}^{N_P} [(1-p_c)(1-p_w)(1-p_d)]^{j\times C_P}$$
$$+ [(1-p_c)(1-p_w)(1-p_d)]^{C_I} \times [(1-p_c)(1-p_w)(1-p_d)]^{N_P \times C_P} \} \times N_{GOP}$$

3 Experimental Settings

In order to evaluate the MPEG video quality using the Q metric, we implemented three connecting simulation interfaces, namely MyTrafficTrace, MyUDP, and MyUDPSink, as well as the random uniform error model, and GE error model [10] into the NS-2 simulator [11]. The MyTrafficTrace extracts the frame type and size from a video trace file, fragments each frame into small packets and forwards these packets to the lower layer at the assigned time. The MyUDP is an extension of the UDP agent that records the timestamp of each of the transmitted packet, the packet id, and the packet payload size in a sender trace file. The MyUDPSink is the receiving agent for the fragmented video frame packets. It records in a receiver trace file similar information as those found in the sender trace file. The Q metric is then computed based on the comparison of these trace files.

The schematic simulation topology is shown in Figure 1. The video server transmits packets over Internet-like and wireless links to reach the video receivers. In our experiments, we adopt the commonly used StarWarsIV and Highway video traffic traces to evaluate the delivered video quality over wireless channels. StarWarsIV is a publicly available long video traffic trace [12], comprising 7500 I frames, 22500 P frames, and 59998 B frames, while the Highway trace is obtained by encoding raw YUV video sequence [13] with NCTU codec [14], including 223 I frames, 445 P frames, and 1332 B frames. The maximum packet sizes are set to 400 and 800 bytes for different packet size comparison. Therefore, C_I would be 4.77 and 9.06, C_P 2.44 and 4.38, and C_B 1.81 and 3.13 for the maximum packet size of 400 and 800 bytes cases for StarWarsIV, respectively. Further, C_I would be 13.01 and 6.73, C_P 4.12 and 2.24, and C_B 3.65 and 2.1 respectively for Highway video sequence.

For the sake of simplicity and to focus on the wireless link evaluation, two assumptions are made. First, infinite play-out buffer is considered to compensate the jitter effect for streaming MPEG video. Therefore, there are no discards due to jitter effects rendering packets too late for being timely played out, i.e. p_d=0. Second, no packet is lost at the wired link between the video server and the base station for the first four experiments, i.e. p_c =0. Each simulation sample results from 30 runs with different seeds and is presented with a 95% confidence interval.

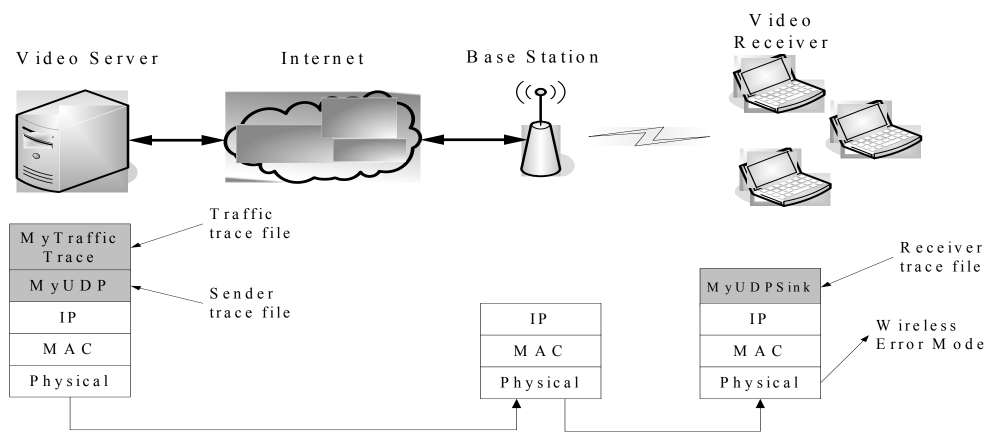


Figure 1 The schematic simulation topology.

4 Simulation Results

In the first experiment, the video packets are delivered via multicast over a random uniform error model of wireless channels. The packet error rate (p) ranges from 0 to 0.2 with 0.02 intervals [15]. Due to the absence of retransmissions in broadcasting and multicasting, the packet error rate at the physical-level is the same as at the application-level. As shown in Figures 3(a-d), as expected, the smaller the packet error rate, the better the evaluated delivery quality by the Q metric of the video flow, regardless of it being the StarWarsIV or Highway traces. Comparing the Q values for different maximum packet sizes, we observe that a smaller packet size results in a worse Q value. For example, when the effective packet error rate is set to 0.02, the average simulated Q values are 0.65 and 0.79 respectively for 400 and 800 bytes cases for Highway. This is because a video frame must be fragmented into more packets for smaller maximum packet size to transmit. The more video packets the source sends, the lower the probability for the receiver to receive all frame packets correctly.

In the second experiment, the video packets are delivered via multicast over wireless GE error model channels. Figure 2 illustrates a state diagram for a Gilbert-Elliott channel model. In the "good" state (G) errors occur with lower probability P_G while in the "bad" state (B) they happen with higher probability P_B. Also, P_{GB} is the probability of the state transiting from a good state to a bad state, and P_{BG} is the transition from a bad state to a good state. The steady state probabilities of being in states G and B are

$$\pi_G = \frac{p_{BG}}{p_{BG} + p_{GB}} \quad \text{and} \quad \pi_B = \frac{p_{GB}}{p_{BG} + p_{GB}},$$

respectively. The average packet error rate produced by the GE error model is

$$p_{avg} = p_G \pi_G + p_B \pi_B$$

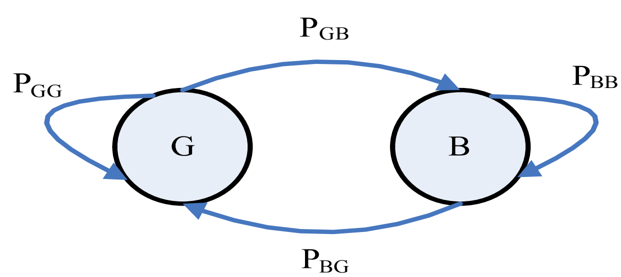
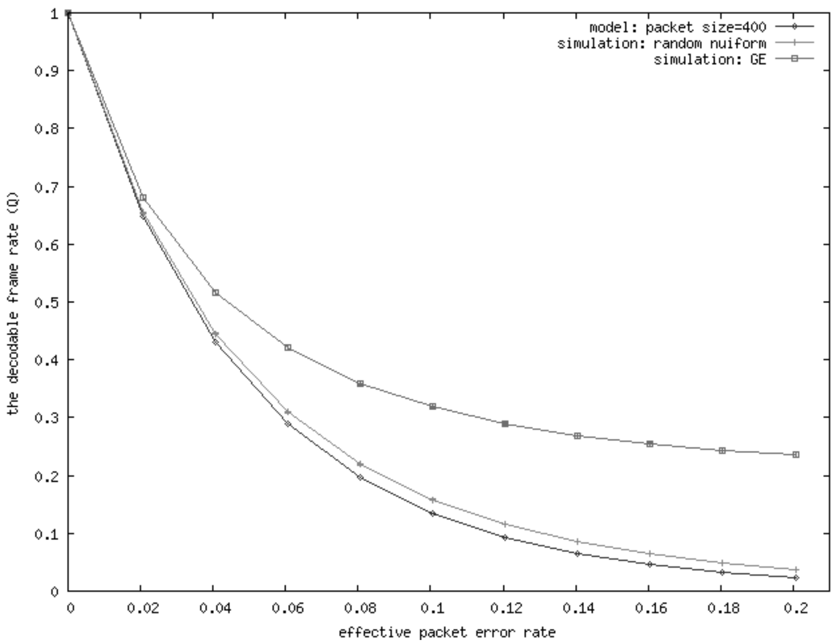
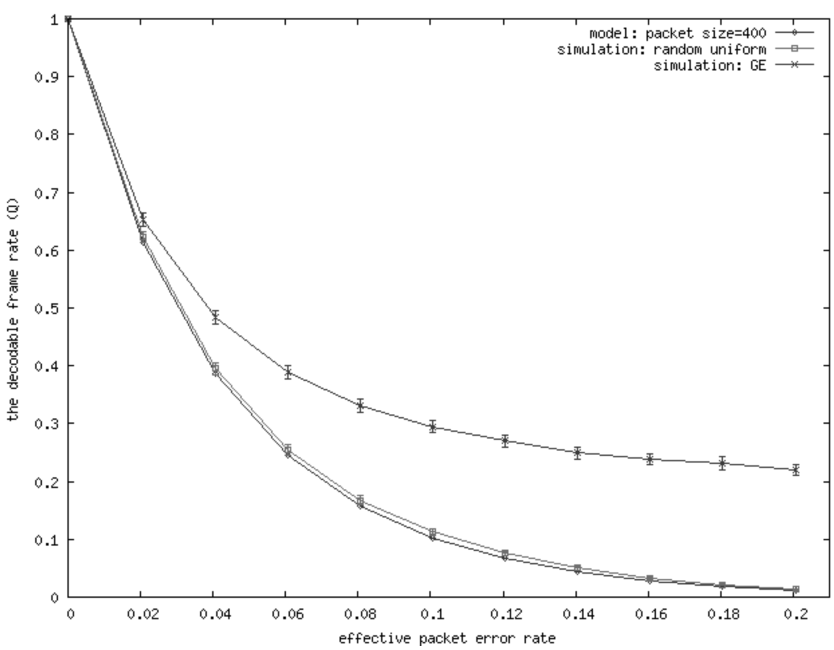


Figure 2 Gilbert-Elliott channel model.

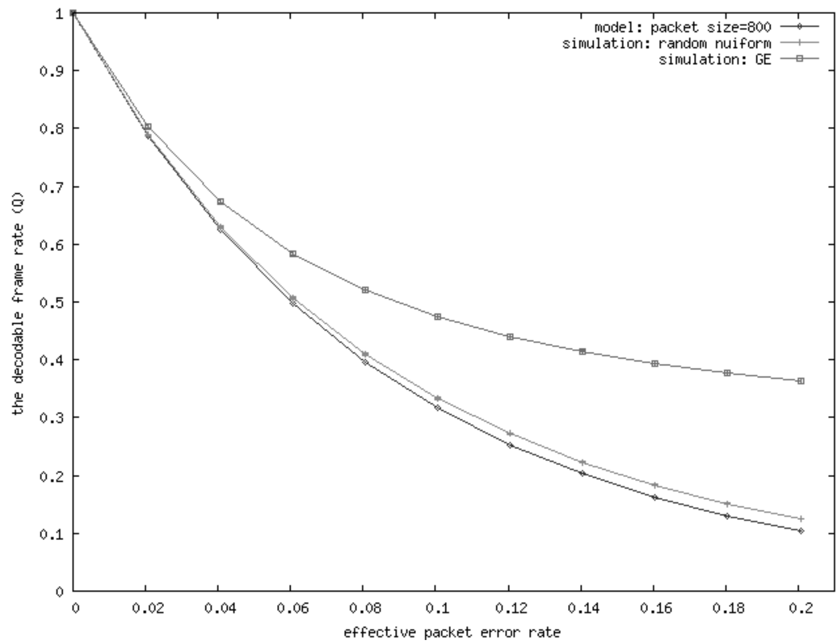
The P_{GG}, P_{BB}, and P_G are set to 0.96, 0.94, and 0.001, respectively. The P_B is set from 0.05 to 0.5 with 0.05 intervals. Therefore, the corresponding effective packet error rates seen at application-level are 0.02, 0.04, …, 0.2 in accordance with the formula shown above. From the results shown in Figures 3(a-d), we observe that when packet error rate is low, the Q metric of the analytical model and simulation are matching. As the packet error rate increases, however, the analytical model underestimates the Q metric performance for the GE error model because this model has the characteristic of a bursty error pattern, thus leading to a lower frame error rate. This phenomenon is illustrated in Figure 4. The packet error rate for both models is 33%, but the corresponding frame error rate of random uniform model (100%) is higher than that of GE model (50%). Therefore, for the same effective packet error rate, the Q metric of simulation provides a lower bound on quality for wireless channels modeled by the GE error model.



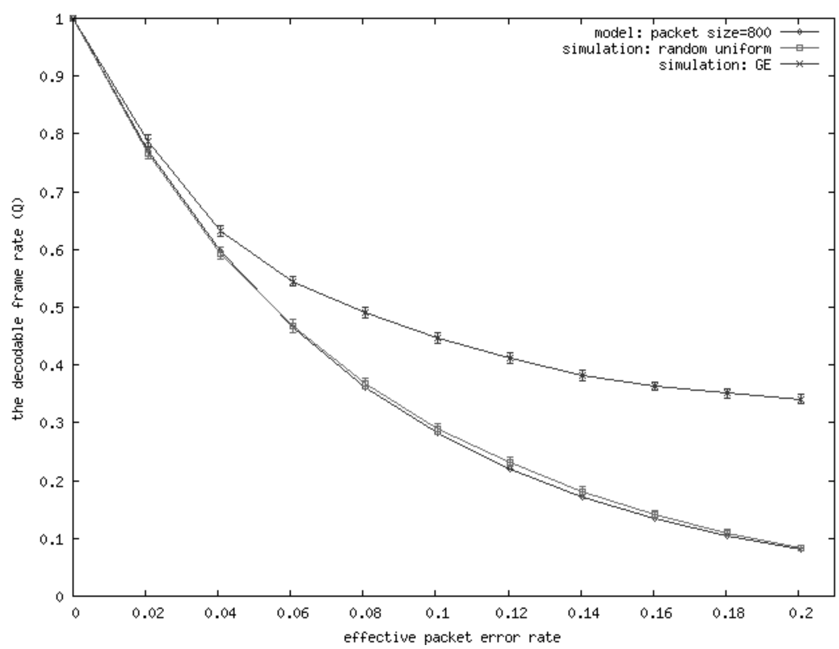
(a) packet size = 400 bytes (StarWarsIV)



(b) packet size = 400 bytes (Highway)



(c) packet size = 800 bytes (StarWarsIV)



(d) packet size = 800 bytes (Highway)

Figure 3 Multicast over random uniform and GE error model channels (Q).

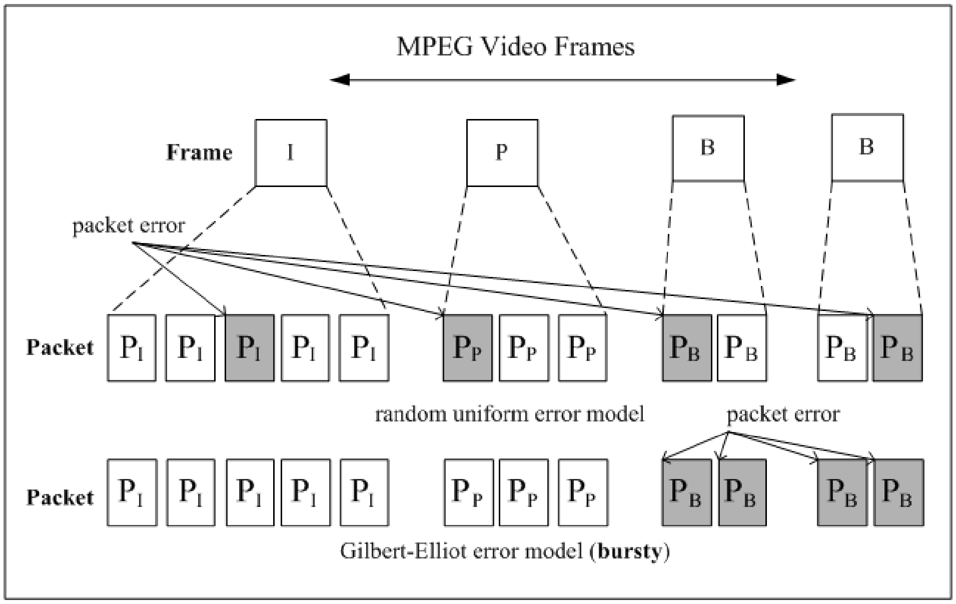


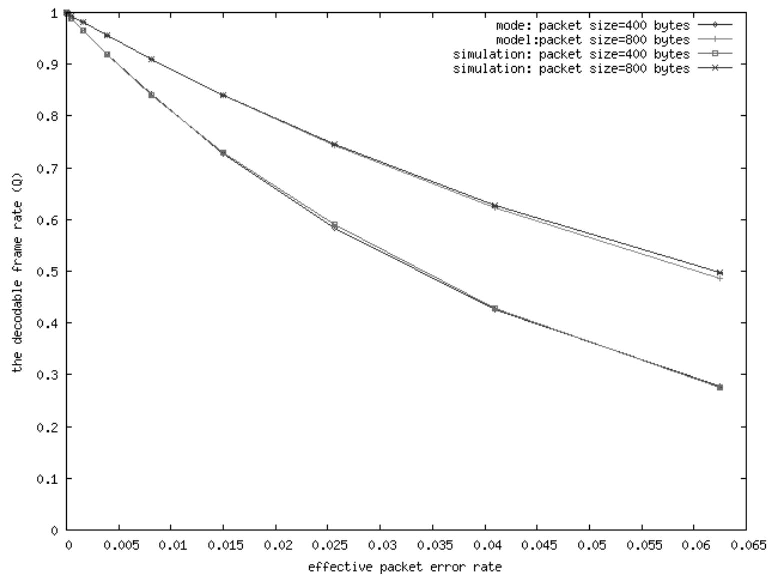
Figure 4 MPEG video transmission with two kinds of wireless error models (Q).

In the third experiment, the video packets are delivered via unicast over a random uniform error model for wireless channels. The packet error rate (p) ranges from 0.05 to 0.5 with 0.05 intervals. Nevertheless, in unicasting, the MAC layer at a sender retransmits an unacknowledged packet at a maximum of N times before it gives up. The perceived correct rate at application-level is thus

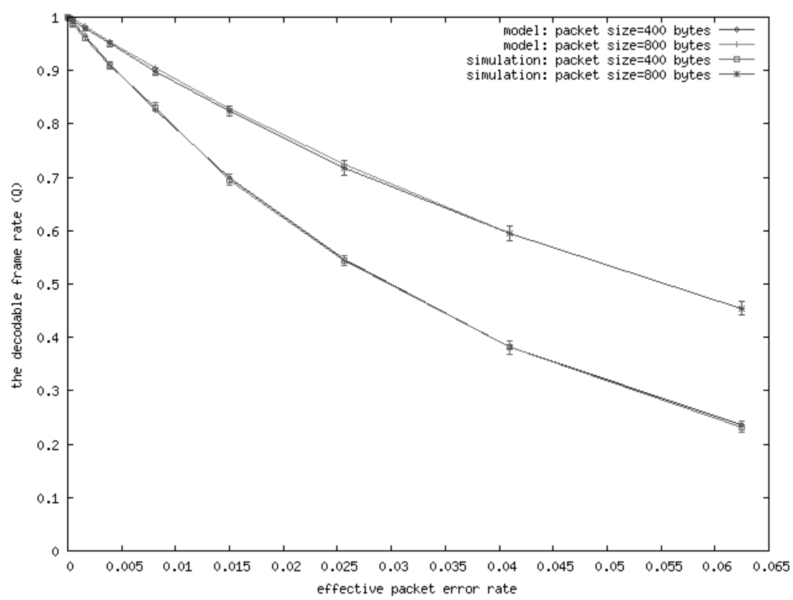
$$p_{CORECT} = \sum_{i=1}^{N}(1-p)p^{i-1} = 1-p^{N},$$

where N is the maximum number of retransmission at the MAC layer and p is the packet error rate at the physical-level. As a consequence, the application-level error rate is $p_{effective} = p^N$. Here, we assume that the maximum number of retransmissions is four, i.e. N=4. Hence, $p_{effective}$ corresponds to 0.00000625, 0.0001, …, 0.0625. As Figure 5(a) and 5(b) show, the Q metric of the analytical model and simulation are very matching no matter the adopted maximum packet size.

In the fourth experiment, we use the Highway traffic trace to study the end-to-end delay of video delivery over wireless channels. The end-to-end delay is also correlated to display video quality. If a frame arrives later than its defined playback time, the frame becomes useless and will be dropped. For multicasting, the base station will forward the packet only once. So the end-to-end delay keeps almost a constant value for all packet error rates in the physical layer no matter if random uniform or GE error model is used. However, for unicasting, the MAC layer of the base station will retransmit an unacknowledged video packet. Therefore, it is clearly shown in Figure 6 that as the effective packet error increases, the average or maximum end-to-end delay increases as well. This phenomenon can be better understood by analyzing Figures 7(a) and 7(b) that represent the variation of the queue size at the base station. When the packet error rate increases, more video packets are delayed in the queue of the base station waiting for the successful retransmission of erroneous video packets.



(a) StarWarsIV



(b)Highway

Figure 5 Unicast over random uniform error model channels (Q).

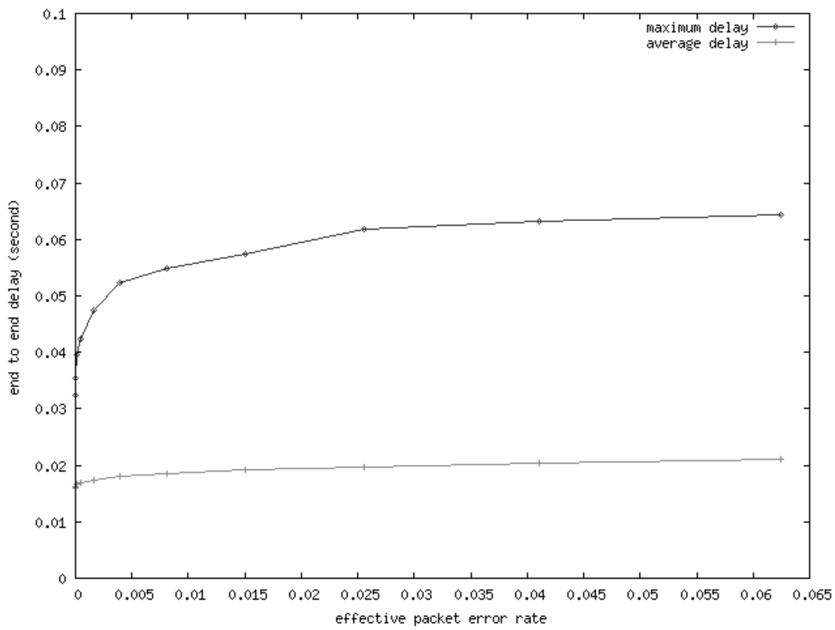
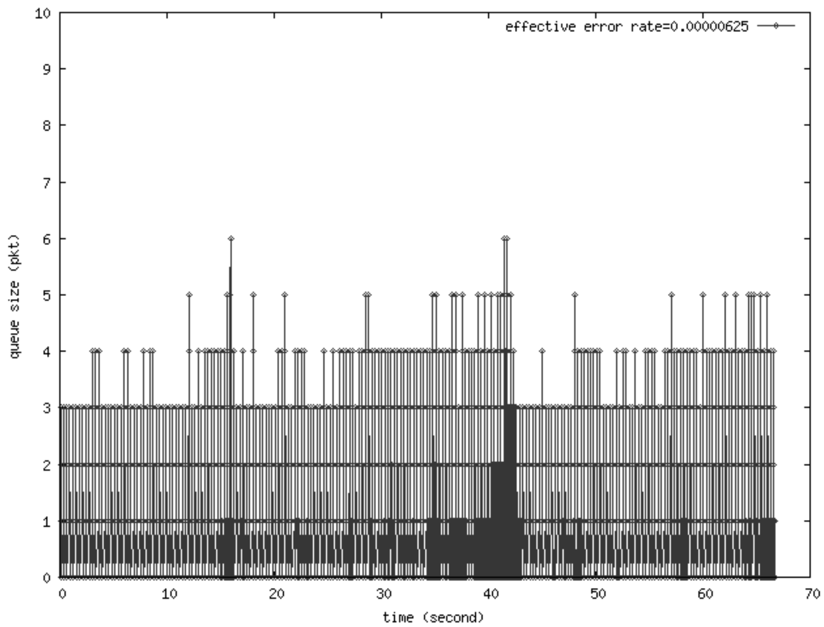
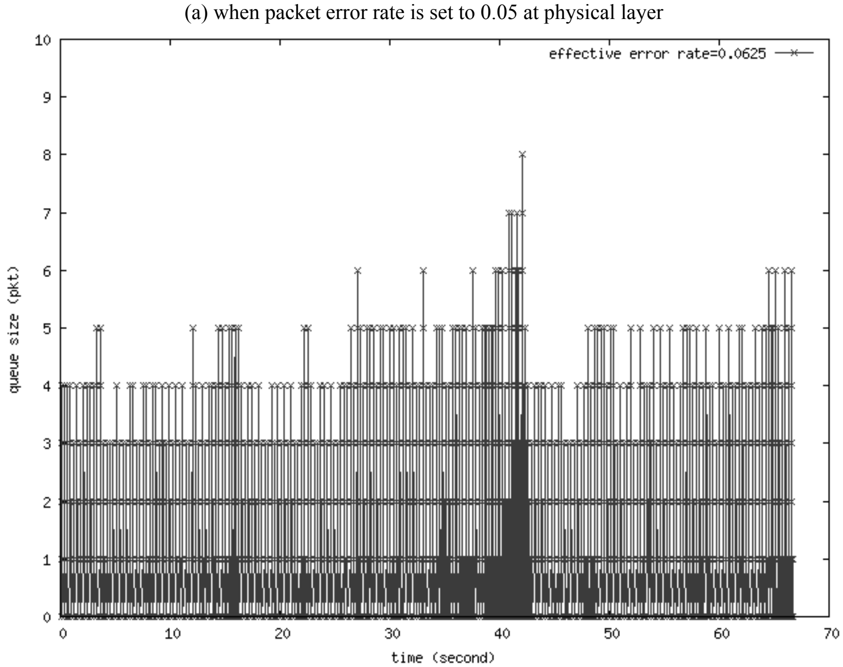


Figure 6 End-to-end delay (for the maximum packet size = 800).

In the fifth experiment, we study the combined effect of the congestion loss and wireless error on the delivered video quality. Although common UDP or TCP implementations do not distinguish congestion losses from wireless losses due to wireless link errors, there is a research trend to handle this issue, such as ECN-based loss differentiation algorithm [16], MAC-based loss differentiation algorithm [17], and inter-arrival time based loss differentiation algorithm [18]. We assume such algorithms are adopted in this paper to differentiate congestion losses from wireless losses. The random uniform error model is inserted in the wired Internet to model congestion loss. The tested StartWarsIV video sequence is transmitted using multicast from wired node to the mobile node. The maximum packet size is 800 bytes. The effective packet error rate is fixed at 0.02 for the random uniform and GE models. The congestion loss rate varies from 0 to 0.1 with 0.01 intervals. The simulation results are shown in Figure 8. The results also show the effectiveness when taking the congestion loss into consideration.



(a) when packet error rate is set to 0.05 at physical layer



(b) when packet error rate is set to 0.5 at physical layer

Figure 7 The queue size variation in the base station.

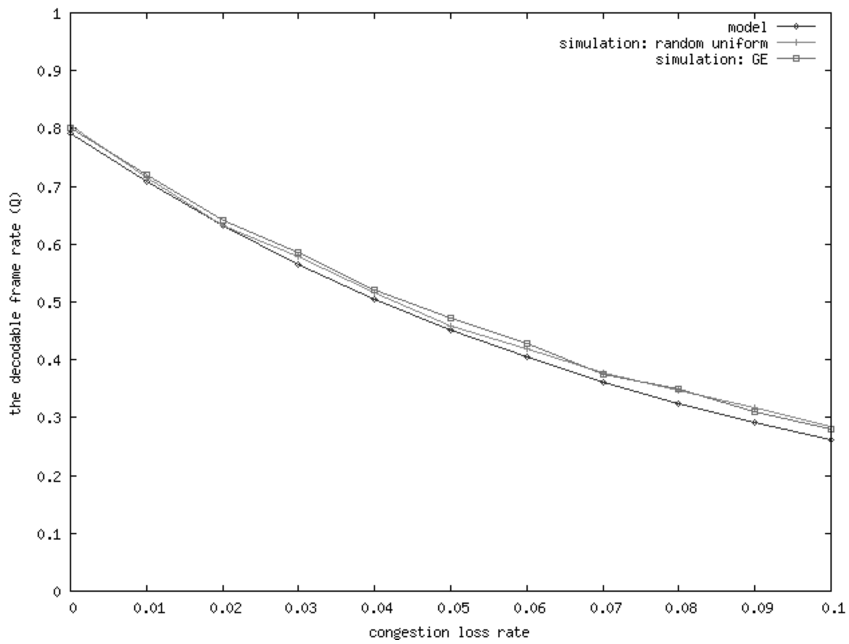


Figure 8 When taking congestion loss into consideration.

5 The Relationship Between PSNR and Q

In this section, we measure PSNR values for the same simulation settings adopted to get results using the Q metric. The evaluation is based on the enhanced Evalvid framework [4, 5], shown in Figure 9. The raw Highway YUV (352 x 288 CIF format) video is encoded by NCTU MPEG4 video encoder [14]. After encoding, video sender (VS program) reads the compressed video file from the output of video encoder, fragments each larger video frame into smaller segments, and then transmits these segments into NS2 environment via UDP packets with the aid of the MyTrafficTrace. After simulation, based on the original encoded video file, the video trace file, the sender trace file, and the receiver trace file, the evaluate trace (ET program) generates a reconstructed video file, which corresponds to the possibly corrupted video found at the receiver side as it would be reproduced to an end user. In principle, the generation of the possibly corrupted video can be regarded as a process of copying the original video trace file frame by frame, omitting frames indicated as lost or corrupted at the receiver side. Then the reconstructed video file is fed into decoder and followed by error concealment. Digital video quality assessment has to be performed frame by frame. Therefore, the total number of video frames at the receiver side, including the erroneous ones, must be the same as that of the original video at the sender side. So we insert the last successfully decoded frame in the place of each lost frame as an error concealment technique (FV program). Finally, the original YUV video can the reconstructed fixed YUV video can be compared to get the PSNR.

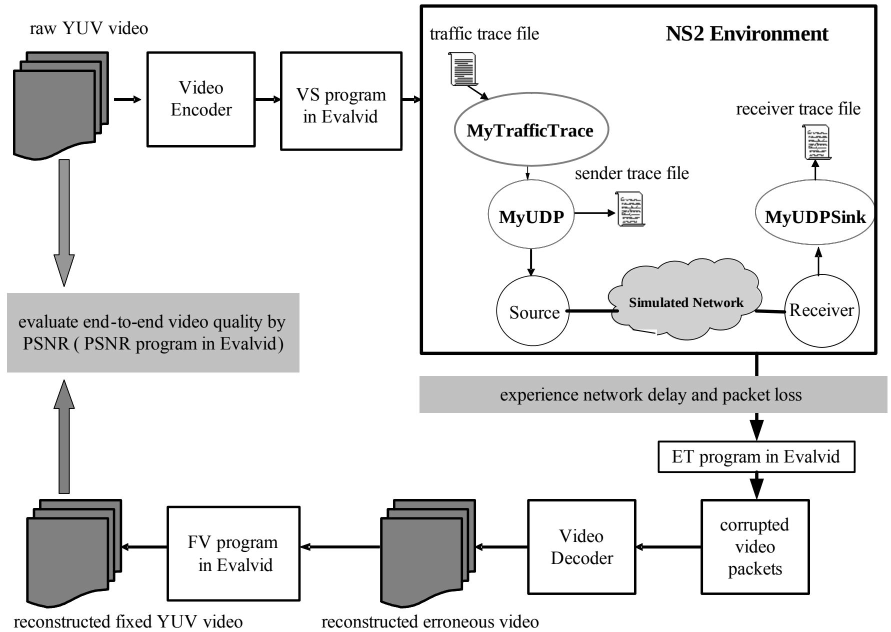
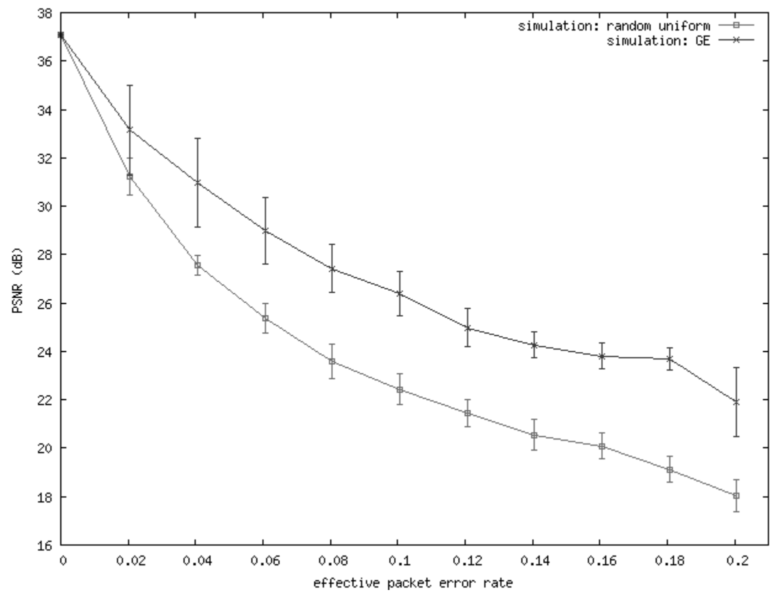


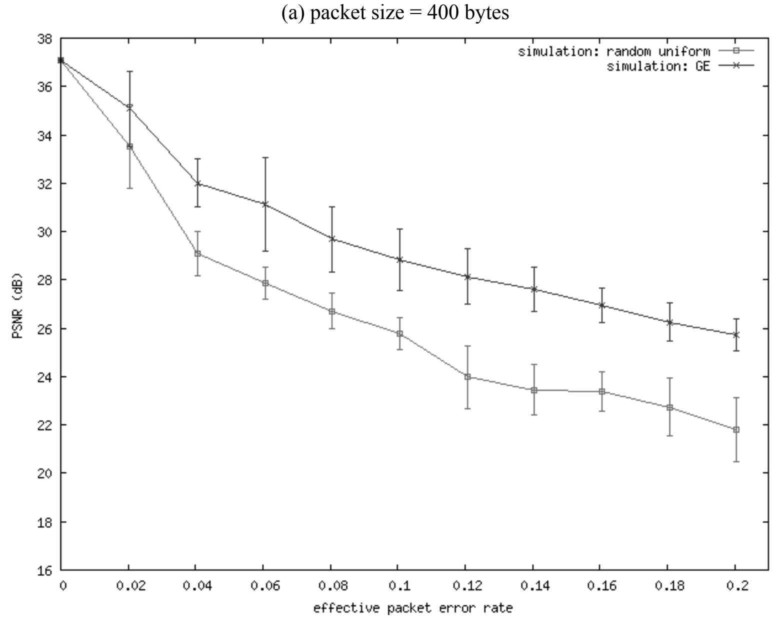
Figure 9 Interfaces between Evalvid and NS2.

The simulation settings are the same as in previous sections. The results for multicasting video packets over wireless random uniform and GE error model channels are shown in Figure 10. Likewise the results for the decodable frame rate, when the effective packet error rate increases, the PSNR decreases. For the same effective packet error rate, the average PSNR value for the maximum packet size is better than the one for a smaller packet size no matter if random uniform or GE error model is used. Further, the average PSNR value for GE model is better than the one for a random uniform considering the same effective rate. The difference between the normalized results of PSNR and Q is within the range of 95% confidence interval. It is clear that the range of 95% confidence interval for PSNR is larger than that for Q, indicating a larger variance in results from the different simulation samples. This is because the enhanced Evalvid system takes the error concealment into consideration and our model does not. When error occurs in these lightly changed frames, the fixed video frames still can get a reasonable PSNR evaluation. On the contrary, when an error occurs in dramatically changed frames (e.g. in scene changing), the erroneous video frame might not have the error properly concealed by a successfully decoded frame. In this paper, we consider the worst-case PSNR for comparison purposes.

The results for unicasting video packets over wireless random uniform model channels are shown in Figure 11. Likewise, when the effective packet rate is low, the PSNR value is high. For the same effective packet error rate, the PSNR value is higher for longer maximum packet sizes. If we compare the PSNR for the same setting of packet error in the physical layer, we can find the better PSNR evaluation for unicast transmission as compared to multicast. This is because when sending video packets in unicast over 802.11 related wireless, the MAC layer will retransmit a packet detected as lost as opposed to a no retransmission scheme in the case of multicasting.



(a) packet size = 400 bytes



(b) packet size = 800 bytes

Figure 10 Multicast over random uniform and GE error model channels (PSNR).

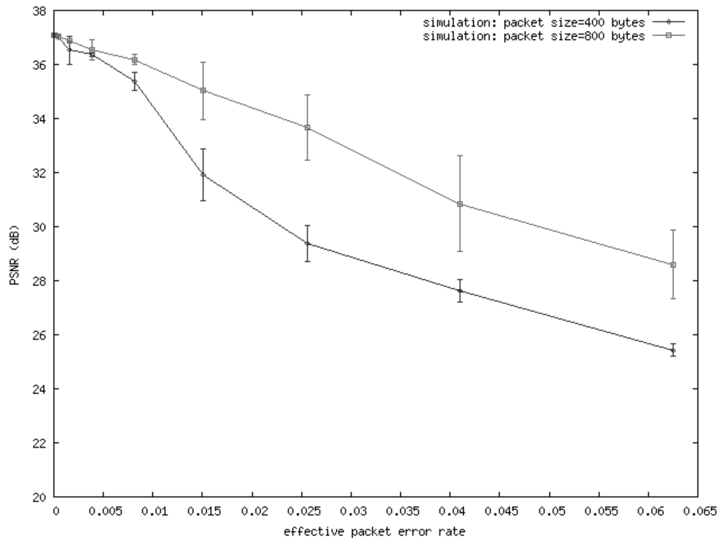
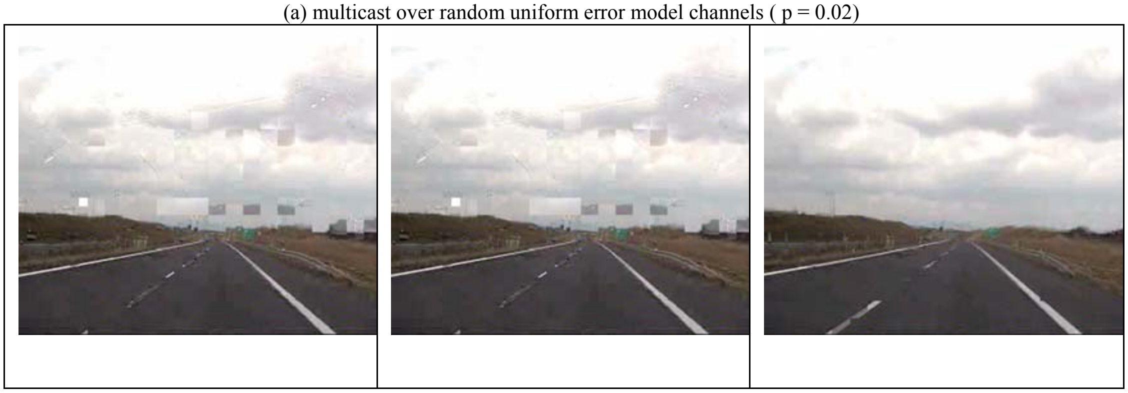


Figure 11 Unicast over random uniform error model channels (PSNR).

(a) multicast over random uniform error model channels (p = 0.02)



(b) multicast over random uniform error model channels (p = 0.12)

(c) multicast over GE error model channels (p = 0.12)

Figure 12 Visual comparison of reconstructed frame.

With the aid of YUVviewer program [19], we can also visually compare the display video quality achieved in our results. Comparing Figures 12(a) and 12(b), we observe that as the effective packet error rate is larger, how worse becomes the delivered quality to video streams. Comparing Figures 12(b) and 12(c), we remark that when effective packet rate is the same, the delivered video quality over GE error model is better than that over random uniform error model.

It is also interesting to have a closer look on this simulation. When computing the PSNR metric, it takes around 3 to 4 minutes to finish the task of simulating, evaluating traces, decoding, fixing, and doing the frame-by-frame PSNR comparison on a Pentium III 1 GHz computer equipped with 512 MB RAM. In contrast, it takes less than 10 seconds to compute the Q metric value representing the fraction of decodable frames. It needs to be carefully noticed that Highway has only 2000 frames or around 1.11 minutes for video transmission at the rate of 30 frames/second. If the test sequence has more frames, more time is needed to finish all the jobs. More formally, the computation complexity to get the Q and PSNR metrics are O(N) and O(kN), respectively, where N is the number of frames in the video sequence and k is the number of pixels within each frame. Therefore, the decodable frame rate metric can adequately reflect the behavior of the PSNR QoS video assessment metric with reasonable accuracy, while being much less time-consuming.

6 Summary and Outlook

In this paper, we analyzed the impact of the loss models on the streaming MPEG video quality by means of an analytical model and extensive simulations. The simulation results show the effectiveness of our analytical model for the uniform random error model. Although there is a deviation for GE error model when packet error rate is higher, the analytical model provides the lower bounds on the performance of delivered MPEG video quality over bursty error model channels. Further, the fraction of decodable frames reflects well the behavior of the well-accepted PSNR metric, whereas being much less time-consuming.

Future direction for this research is to improve the accuracy of the analytical model while considering burst errors as those provided by the Gilbert-Elliot model.

References

1. M. D. de Amorim, O. C. M. B. Duarte, and G. Pujolle, Distinguishing video quality through differential matrices, ACM Multimedia Systems Journal, vol. 9, no. 1, pp. 94–103, 2003.
2. J. Shin, J. Kim, and C. C. J.Kuo, Quality of service mapping mechanism for packet video in differentiated services network, IEEE Transactions on Multimedia, vol. 3, no. 2, pp. 219–231, June 2001.
3. A. Ziviani, B. E. Wolfinger, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, Joint adoption of QoS schemes for MPEG streams, Multimedia Tools and Applications, vol. 26, no. 1, pp. 59–80, May 2005.
4. J. Klaue, B. Rathke, and A. Wolisz, EvalVid - A Framework for Video Transmission and Quality Evaluation, In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, USA, Sept. 2003.
5. http://hpds.ee.ncku.edu.tw/~smallko/ns2/Evalvid_in_NS2.htm.
6. C. H. Ke, C. K. Shieh, W. S. Hwang, and A. Ziviani, A two-markers system for improved MPEG video delivery in a diffserv network, IEEE Communications Letters, vol. 9, no. 4, pp. 381–383, Apr. 2005.
7. J. Naoum-Sawaya, B. Ghaddar, S. Khawam, H. Safa, H. Artail, and Z. Dawy, Adaptive Approach for QoS Support in IEEE 802.11e Wireless LAN, in IEEE International Conference on Wireless and Mobile Computing , Networking and Communications, Montreal, Canada, Aug. 2005.
8. H. Huang, J. Ou, and D. Zhang, Efficient Multimedia Transmission in Mobile Network by using PR-SCTP, Communications and Computer Networks, Marina del Rey, USA, Oct. 2005.
9. Tudor, P.N, MPEG-2 video compression tutorial, MPEG-2 - What it is and What it isn't, IEE Colloquium on, 24 Jan. 1995.
10. E. N. Gilbert, Capacity of a burst-noise channel, Bell System Technical Journal, vol. 39, pp. 1253–1266, Sept. 1960, monograph 3683.
11. The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/.
12. http://www.tkn.tu-berlin.de/research/trace/pics/FrameTrace/mp4/Verbose_StarWarsIV.dat.
13. YUV video sequences (CIF), http://www.tkn.tu-berlin.de/research/evalvid/cif.html.
14. NCTU codec, http://megaera.ee.nctu.edu.tw/mpeg.
15. J. H. Choi, J. G. Choi and C. Yoo, Adapting TCP Segment Size in Cellular Networks, International Conference on Networking, ReunionIsland, France, April 17-21, 2005.
16. S. Bae, and S. Chong. TCP-Friendly Wireless Multimedia Flow Control Using ECN Marking. IEEE GLOGALCOM, vol. 2, pp. 1794-1799, Nov. 2002.
17. B. Zhang, and M. N. Shirazi, Implementation of Explicit Wireless Loss Notification Using MAC-Layer Information, Wireless Communications and Networking, vol. 2, pp. 1339-1343, Mar. 2003.
18. S. Cen, P. C. Cosman, G. M. Voelker, End-to-End Differentiation of Congestion and Wireless Losses, IEEE/ACM Transactions on Networking, vol. 11, pp. 703-717, Oct. 2003.
19. yuvviewer, http://eeweb.poly.edu/~yao/VideobookSampleData/video/application/YUVviewer.exe.