Journal of Engineering Science and Technology
Vol. 7, No. 3 (2012) 402 - 417

A PERFORMANCE STUDY OF ADAPTIVE FEC MECHANISM FOR VIDEO TRANSMISSION OVER 802.11 WLANS

NUR ZIADAH HARUN[1,*], OSMAN GHAZALI[2]

[1]School of Computing, University Utara Malaysia, 06010 Sintok, Kedah, Malaysia
[2]College of Arts and Sciences, University Utara Malaysia, 06010 Sintok, Kedah, Malaysia
*Corresponding Author: s71497@student.uum.edu.my

Abstract

Forward Error Correction has been implemented with Automatic Repeat reQuest to overcome packet losses and avoid network congestion in various wireless network conditions. The number of FEC packets is needed to be generated adaptively due to various network states and burst error condition. There are a number of proposed Adaptive FEC mechanisms that can generate FEC packets adaptively for wireless network. However, this current Adaptive FEC, namely Mend FEC mechanism has major drawbacks including injecting excessive number of FEC packets into the network and consequently reducing recovery performance. This paper proposed an enhancement on Mend FEC mechanism, which is called Enhanced Adaptive FEC (EnAFEC) mechanism. The enhancement is aimed to improve recovery performance of the Mend FEC mechanism by generating FEC packets dynamically based on varying wireless network conditions. The proposed enhancement was implemented in simulation environment using the NS2 network simulation. The results show that EnAFEC mechanism outperformed the other Adaptive FEC mechanism in terms of recovery efficiency. This paper also highlights that even minimal amount of FEC packets can recover high packet losses and achieve good video quality at the receiver.

Keywords: Forward error correction, Automatic repeat reQuest, Smoothing factor, Block length adaptation, Burst error, Wireless network.

1. Introduction

Video transmission over the wireless network is usually interrupted by video packet loss that is caused by interference, terrestrial obstructions, and reflection of transmission signal [1]. To ensure that the video delivered at the receiver end is in good quality, Forward Error Correction (FEC) can be used to recover the video packet from being lost. The principle of FEC is to add redundant packet so that

Nomenclatures

avg_q	Average queue length
avg_{rT}	Average retransmission times
fecpkts	Number of FEC packets
$inst_q$	Current queue length
$inst_{rT}$	Current retransmission times
P_B	The probability that the channel is in bad status
P_{BB}	The probability that the channel is kept in bad status
P_G	The probability that the channel is in good status
P_{GG}	The probability that the channel is kept in good status
qlen	Queue length
rT	Retransmission times
w_q	Smoothing factor for queue length
w_{rT}	Smoothing factor for retransmission

Greek Symbols

α	Weight to determine final number of FEC packets

Abbreviations

ARQ	Automatic Repeat reQuest
EnAFEC	Enhanced Adaptive FEC
FEC	Forward Error Correction
PSNR	Peak-Signal-to-Noise Ratio

the original packet can be reconstructed in the occurrence of packet loss. In order to generate the appropriate number of FEC, Automatic Repeat reQuest (ARQ) mechanism can be adopted with FEC mechanism. The adaptation is necessary to handle the various wireless network conditions, as each mobile node experiences different channel conditions. This leads to the difficulties in deciding the correct number of FEC packets to be generated. Small numbers of redundant packets lead to small overhead, which might not be able to recover all loss packets. As a result, a low video quality is produced. On the other hand, large numbers of redundant packets produce large overhead and in good video quality [2]. However, generating large number of redundant packets wastes network bandwidth which also contributes to network congestion.

In wireless network, queue length at Access Point is used as the indicator for estimating network traffic load. Several works proposed the dynamic FEC mechanism based on the average packet queue length to ensure the FEC packets do not congest the network. For example, the Enhanced Adaptive Forward Error Correction (EAFEC) mechanism proposed by Lin et al. in [3] has implemented dynamic FEC mechanism at the Access Point. This mechanism generates the FEC packets according to the network status. Meanwhile, Du et al. in [4] proposed Mend FEC which is an enhancement from EAFEC mechanism that can improve the quality of video in sudden video changing scene. In EAFEC mechanism, when queue length is too large, video packets will be transmitted without adding FEC packets. This is due to the fact that if queue length is more than the certain threshold, the number of FEC is set to zero. If the wireless channel state is worst at that time, the original packets might be dropped and the receiver will not be able to recover the packets. Thus, retransmission time in EAFEC is not a good indicator to estimate the number of FEC as it does not fully adapt to the various wireless network conditions.

EAFEC [3] and Mend FEC [4] use uniform error model for verification purposes since the model is easier to implement compared to the Gilbert-Elliot model. However, the uniform error model is unable to represent the burst error network that usually occurs in the wireless network. Therefore, Adaptive, Hybrid ARQ and FEC (AHAFEC) [5] mechanism that are able to alter the amount of FEC packets and the number of maximum retransmission at the Access Point are proposed. The performance of AHAFEC is better than EAFEC by generating low number of loss frame. However, AHAFEC has a limitation whereby high number of retransmission time will consume delay. Apart from that, these three adaptive FEC mechanisms use average queue length when generating FEC packets. Unfortunately, none of them estimates the suitable smoothing factor value to determine the average queue length.

The aim of EnAFEC mechanism is to improve the performance of the existing Adaptive FEC mechanism, which is Mend FEC [4] in terms of recovery efficiency. The combination techniques on block length adaptation and suitable smoothing factor value are introduced. To evaluate the performance of EnAFEC, performance metrics such as peak-signal-to-noise ratio PSNR, recovery efficiency and FEC efficiency have been used. The next section discusses the enhancement of the Mend FEC mechanism. Then, the simulation topology and its setting are introduced. The simulation results from the experiments are also discussed as they provide the evidence that EnAFEC is better than the EAFEC and Mend FEC. Lastly, the conclusions are given based on the performance study from the experimental results.

2. Background of Adaptive FEC Mechanism

In the wireless network, there are two common approaches used to recover packet error. The approaches are Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ). The aim of FEC is to add redundant information to a video stream so that it can be used to recover the lost packet. FEC is a technique for error correction over the network that is used by the receiver without requiring retransmission of any information from the sender.

Figure 1 outlines the basic operation of the FEC mechanism. In the FEC approach, every K video packet is protected by (N-K) FEC packet. The first K packet of the N group is the original data packet while the remaining (N-K) is known as parity data [6]. To make sure that the original data is recovered, K out of the N unit must be received at the receiver. Based on Fig. 1, as long as K out of N unit is received, the original data packets will be successfully recovered from loss due to the lossy network.

ARQ is a mechanism to detect packet losses, which operates at the MAC layer. The use of ARQ with FEC mechanism is to determine the number of FEC packets to be added to the original packets. It is based on packet retransmission times. The ARQ mechanism provides reliable data transfer. ARQ operates with two ways of activating mechanism. Firstly, upon request from the receiver and secondly is upon timeout of the timer at the sender [7].

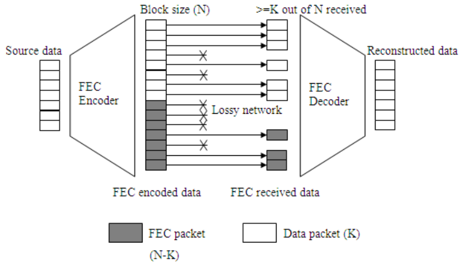

Fig. 1. Basic Operation of FEC Mechanism [8].

The operation of ARQ mechanism is shown in Fig. 2. Generally, the standard IEEE 802.11 WLAN mechanism defines that a packet will be kept in the sending buffer until an ACK packet is received or the number of retransmission exceeds a certain threshold (RTS_Threshold) [9]. Thus, for each time the sender does not receive ACK packet, the packet is assumed to be failed during transmission. This standard consists of four-way information exchange including Ready-to-Send (RTS), Clear-to-Send (CTS), Data and Acknowledgement (ACK) [10]. Based on Fig. 2, retransmission occurred when a CTS or an ACK frame is not received after RTS or Data frame transmission [11]. The retransmission is caused by either dropping of a frame or congestion of wireless network. The operation of retransmission is started when the sender sent RTS packet to the receiver. The receiver will then respond with a CTS packet. Then, the sender completes the transmission of data to the receiver. Once each packet is sent out, the sender does not send any further packets until it receives ACK from the receiver. If the packet successfully arrived at the receiver without error, the receiver will respond to the sender with an ACK packet. The packet is assumed to be lost if the ACK packet is not received. Only then, the same packet will be retransmitted to the receiver. If the sender fails to receive the ACK prior to timeout, the same packet will be resend to the receiver. If the retransmission counter reaches the RetryLimit, MAC layer will then report to the upper layer, i.e. the network layer. This means that the packet is not successfully delivered to the receiver. Based on the finding in [12], ARQ mechanism is based on retry count. When the packet failed during transmission, the retry count is increased by one. Otherwise, the packet will be continuously retransmitted until it is discarded if the retry count is greater than the retry limit. The ARQ mechanism is easy to implement and very effective against burst error. However, it consumes delay, requires feedback, and faces scalability issue for multicast transmission [13]. Kennedy [14] performed an average analysis to determine a suitable burst duration limit. It is demonstrated that an optimized CFB configuration allows the MAC protocol to achieve 30% more capacity than the basic EDCA scheme.

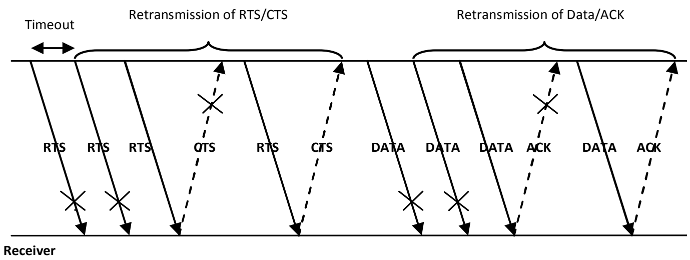

Fig. 2. Operation of ARQ Mechanism [11].

3. Computer Programme: Validation and Verification

In enhanced adaptive FEC mechanism, the number of FEC is generated based on the current network condition as proposed by Lin et al. [3]. Queue length at the access point is a good indicator to estimate network traffic load. On the other hand, packet retransmission times are also used to measure wireless channel status. However, this mechanism has some limitations in generating the number of FEC. When qlen>th2, the number of FEC is set to 0. That means when the queue length is almost full, no FEC packets will be generated. The problem occurs when the wireless channel is worst at the time. Consequently, there is no recovery for packets losses which produce bad video quality at the receiving end.

The limitation of EAFEC mechanism is improved by modifying the way FEC packets generated at the access point as proposed by [4]. The enhancement is made based on the number of FEC packets by calculating separately based on the queue length and packet retransmission times. The steps are described as follows; firstly, if the queue length is smaller than threshold1, the number of FEC is set to the maximum. If it is larger than threshold2, there are no FEC packets generated because the queue length is heavily loaded. Otherwise, the FEC packets are generated based on data size fraction in queue length.

Secondly, the FEC packets are calculated again based on the packet retransmission time. If the retransmission times are less than threshold3, no FEC packets are generated. If the retransmission times are greater than threshold4, FEC packets are set to the maximum as the probability of packets loss increases. Otherwise, the FEC packets increase together with the retransmission times. Lastly, the final redundant packets can be acquired by calculating the set of redundant FEC packets with weight values.

From the Mend FEC algorithm on Fig. 3, it can be seen that the FEC packets are generated along the time. To reduce the unnecessary FEC packets, block length adaptation needs to be implemented with the Mend FEC algorithm. A video frame (or a block) has a fixed number of video packets. In this experiment, a block of packet is considered for the FEC packet generation, whereby the FEC packets are generated on the top of one video block. To minimize network congestion, the maximum number of FEC packets must not exceed the number of original video packets in a block. The preliminary experiment was conducted to determine the appropriate FEC blocks.

In the preliminary experiments, the number of FEC generated with different video block length (8, 10, and 12 packets) under different packet error rates was compared. Block length was determined by the number of video packets per block. FEC blocks were generated on the top of each video block. The results of the experiment are presented in Tables 1, 2 and 3.

Table 1. Block Length Adaptation with Error Probability 0.2.

Block Length	No. of FEC blocks	No. of FEC packets	Recovery efficiency	FEC efficiency
8 packets	363±6	363×8=2904	0.0022	0.930
10 packets	284±14	284×10=2840	0.0050	0.945
12 packets	83±5	83×12=996	0.0053	0.987

Table 2. Block Length Adaptation with Error Probability 0.3.

Block Length	No. of FEC blocks	No. of FEC packets	Recovery efficiency	FEC efficiency
8 packets	422±18	422×8=3376	0.011	0.920
10 packets	357±10	357×10=3570	0.020	0.932
12 packets	150±9	150×12=1800	0.025	0.971

Table 3. Block Length Adaptation with Error Probability 0.4.

Block Length	No. of FEC blocks	No. of FEC packets	Recovery efficiency	FEC efficiency
8 packets	576±14	576×8=4608	0.037	0.897
10 packets	483±24	483×10=4830	0.049	0.913
12 packets	251±8	251×12=3012	0.070	0.954

Initialization:
qlen = 0; rT = 0;

When a block of packet arrives:
/* use queue length to determine number of FEC packets */
$avg_q = (1 - w_q) \times inst_q + w_q \times avg_q$
if (avg_q < th1)
num_FEC1 = Max_FEC;
else if (avg_q < th2)
num_FEC1 = Max_FEC * (th2 – avg_q) / (th2 – th1);
else
num_FEC 1= 0;

/* use retransmission times to determine number of FEC packets again */
$avg_{rT} = (1 - w_{rT}) \times inst_{rT} + w_{rT} \times avg_{rT}$
if ($avgrT$ > th4)
num_FEC = Max_FEC;
else if ($avgrT$ < th3)
num_FEC = Max_FEC * (1 – ((th4 – avg_{rT}) / (th4 –th3)));
else
no_FEC = 0;

/* calculate final number of FEC packets */
$fecpkts = \alpha \times num\_fec1 + (1 - \alpha) \times num\_fec2$

Fig. 3. Mend FEC Pseudo Code.

Figure 4 shows the number of FEC packets for different block length adaptations under different error probabilities. It can be seen from this figure that 12 packets per block contribute less FEC packets generation. The reason for this result is that the longer the block length, the lower the number of redundant packets transmitted into the transmission channel. To determine the number of FEC packets generated, the block length must be multiplied by the number of FEC blocks (block length × No. of FEC blocks = No. of FEC packets). As the error probability increases, the FEC packets would also be increased in order to recover more packets lost.

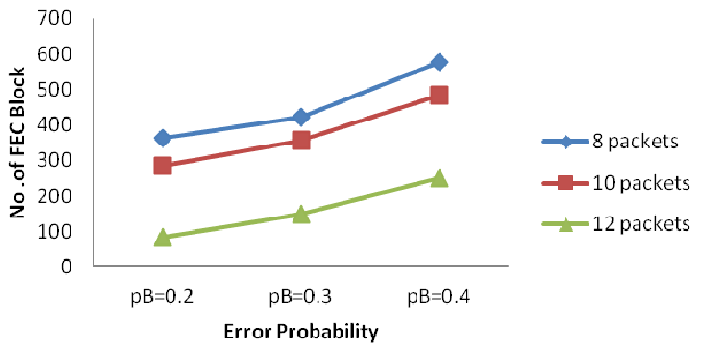

Fig. 4. Number of FEC Block for Different Block Lengths.

Figure 5 shows the recovery efficiency for the different block length adaptation. It can be seen that the 12 packets per block have higher recovery efficiency. It is because a large FEC block length will enhance the recovery performance and reduce the packet error rate [15].

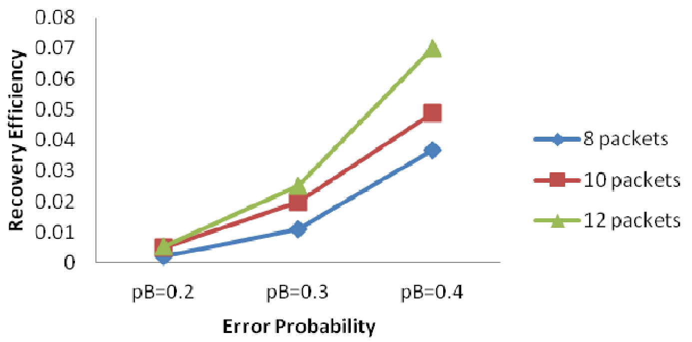

Fig. 5. Recovery Efficiency for Different Block Lengths.

Figure 6 shows the FEC efficiency for the different block length adaptation. It shows that using 12 packets per block produces higher FEC efficiency compared to other block length adaptations. For the next experiment, 12 packets per block will be used due to its recovery ability.

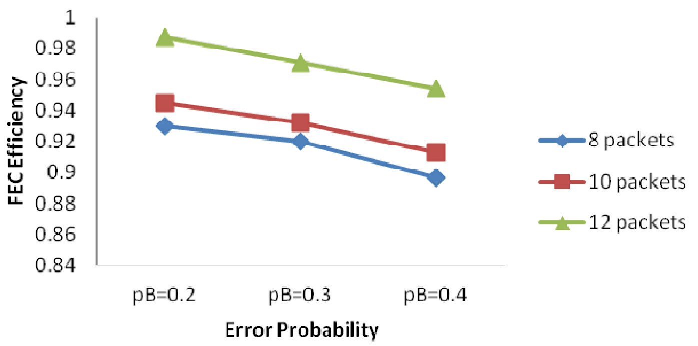

Fig. 6. FEC Efficiency for Different Block Lengths.

4. Enhancement on Smoothing Factor value in Queue Length

The existing Adaptive FEC, which are EAFEC and Mend FEC uses Exponential Weighted Moving Average (EWMA) to estimate the value of average queue length. EWMA is used to minimize the bias against transient burst in the queue length. Whenever the packets queue in the buffer, the average queue length is updated according to the Eq. (1):

$$avg_q = (1 - w_q) \times inst_q - w_q \times avg_q \qquad (1)$$

where $avgq$ is the average queue length, w_q is the smoothing factor and $inst_q$ is the current queue.

Smoothing factor is implemented in EWMA to produce weight average values in order to eliminate the effect of short term fluctuation over the traffic pattern [16]. Based on Eq. (1), w_q, the smoothing factor acts as an important role in determining the queue size used in the averaging process [17]. w_q is set with static value in the range of 0 to 1 to determine the average queue length. Greater value of w_q will produce the best video quality, i.e., 0.9 when the wireless error rate is low [18]. Otherwise, when the wireless conditions become worse, the w_q needs to be set to minimal value, i.e. 0.1, so that more redundant packets can be generated. Therefore, w_q gives the important impact in determining the appropriate average queue length. Its relationship with video quality at the receiver cannot be ignored.

The appropriate values of smoothing factor can be generated based on the number of packet retransmission times at the MAC layer. When the packet retransmission time is low, the value of smoothing factor must be set to a high value. The number of FEC packets generated will be low as the error rate is low. On the other hand, as the packet retransmission time increases, the value of smoothing factor must be decreased. The decrease of this value will result in the ability to generate more FEC packets to recover the failed packet due to the increased of error rates. Here, the new values of wq can be generated as:

if (avg_{rT} < th3)
  w_q = 0.5;
else if (avg_{rT} < th4)
  w_q = int (0.5*(th4 – avg_{rT})/th4-th3);
else
  w_q = 0.1;

Denotes that avg_{rT} is the average of packet retransmission times during retransmission at the MAC layer. Th3 is the low threshold and th4 is the high threshold value for the number of packet retransmission times. When avg_{rT} is less than the certain threshold (th3), the value of w_q is set to 0.5. If the avg_{rT} is larger than th4 value, the value of w_q is set to 0.1. The value of w_q decreases based on the increase of packet retransmission time.

5. Simulation Topology and Setting

The simulation topology in this paper is shown in Fig. 7. In this simulation, video server transmits video streams over the Internet using wired link while wireless nodes are connected using wireless link. The video traffic trace used for this experiment is

"Highway" video using H.264 video coding with JM 1.7 codec. JM 1.7 is used in this experiment because the newer version of JM does not support packet losses [19]. This means that when some parts of the compressed file have been removed or if those packets are lost during transmission, the distorted file is unable to be decoded. Thus JM 1.7 is used to encode and decode the video sequence. The "Highway" video format is Quarter Common Intermediate Format (QCIF) and the Group of Picture (GOP) structure is IPPPPPPPPPPPPPP (Simple Profile). "Highway" video trace consists of 2000 frames, which are divided into transmitting slices. Each slice is about 500 bytes and transmitted via multicast transmission with the GE error model. The P_{GG}, P_{BB}, and P_G are set at 0.96, 0.94, and 0.001 respectively. The packet error probability (P_B) represents the channel that is in a bad state, which varies from 0.1 to 0.5, with 0.1 intervals. The frame rate for this video is 30 (frame/sec) and the total video packets sent are 4829. There are two background traffics in this simulation. The first is the FTP traffic that is transmitted using TCP packets. The second is the exponential traffic transmitted using UDP packets. The transmission rate for the traffic is 1 Mbps which include burst and idle time. Both are set at 0.5ms. The link between wireless AP and the wireless node is IEEE 802.11b 11 Mbps while the link between Internet and wireless AP is 100Mbps. The link between Internet and each traffic source is set at 10 Mbps.

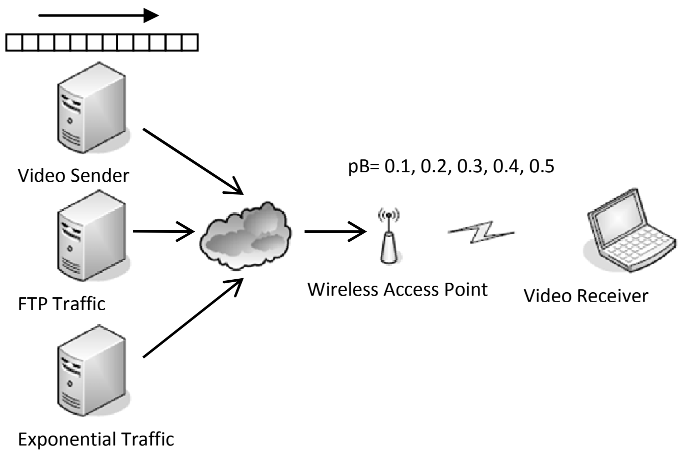

Fig. 7. Simulation Topology for Experiments.

6. Simulation Results and Discussion

This section discusses the simulation results obtained from the performance evaluation on the EAFEC, Mend FEC and EnAFEC mechanisms. All the results were generated from repeated simulations with different packet error rates ranging from good network condition (0.1) to bad network condition (0.7).

As shown in Table 4 and Fig. 8, the numbers of FEC packets increases as the packet error rate increases. This is because high packet error rates leads to high packet retransmission until the packet has been correctly received at the receiver. As the packet error increases, the Mend FEC generates the largest number of FEC packets followed by EAFEC and EnAFEC. Even when the packet error rate is low, Mend FEC generates high FEC packets. However, only small numbers of FEC packets are needed to recover the small number of packet losses. By transmitting original data packets with large number of FEC packets, the network will become congested. This situation occurs because the packets have to queue and wait longer in the AP buffer before it can be transmitted to the receiver. Therefore, even though the network condition is good, Mend FEC is not suitable to prevent network congestions caused by the excessive number of FEC packets. In contrary, EnAFEC generates the lowest FEC packets compared to others.

Table 4. Number of FEC.

	EAFEC	EnAFEC	Mend FEC
P_B=0.1	3±1	1±0	55±3
P_B =0.2	14±0	15±1	82±1
P_B =0.3	49±2	57±2	152±3
P_B =0.4	119±5	110±3	244±6
P_B =0.5	195±6	148±4	331±5
P_B =0.6	269±10	168±3	387±6
P_B =0.7	300±10	178±3	429±5

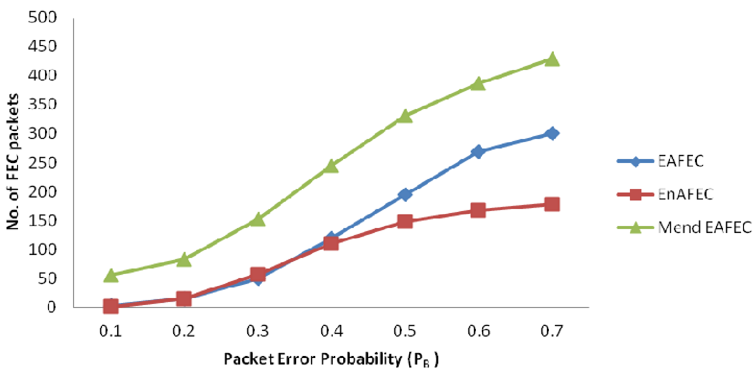

Fig. 8. Number of FEC vs. Error Probability.

Recovery efficiency (RE) is used to measure the ratio of the number of video packets recovered to the total number of FEC packets. Good performance of error recovery would achieve a high value of RE when the value is closer to 1. As shown in Table 5 and Fig. 9, EnAFEC achieves greater RE. In other words, EnAFEC provides a better packet loss recovery performance compared to the other mechanisms. The number of FEC packets generated by EnAFEC is utilized more efficiently to recover the packets lost. The lowest RE value is generated by EAFEC mechanism because it generates more than one FEC block for the same video block. For real video trace file, the missing packet sequence can only be recovered by the same packet sequence generated by FEC. Moreover, Mend FEC was unable to achieve the same recovery performance as EnAFEC even though Mend FEC produces higher number of FEC packets. The excessive FEC packets consume more network bandwidth and waste the resource because not all FEC packets will be used to recover the packets lost.

Table 5. Recovery Efficiency.

	EAFEC	EnAFEC	Mend FEC
P_B=0.1	0±0	0±0	0±0
P_B =0.2	0.0095±0.0049	0.0149±0.009	0.0059±0.0015
P_B =0.3	0.03±0.06	0.05±0.002	0.029±0.005
P_B =0.4	0.0456±0.0071	0.1119±0.0120	0.0609±0.0052
P_B =0.5	0.0935±0.0070	0.1956±0.0165	0.1179±0.0055
P_B =0.6	0.1489±0.0071	0.2851±0.014	0.1772±0.0087
P_B =0.7	0.2063±0.01	0.314±0.012	0.2213±0.0053

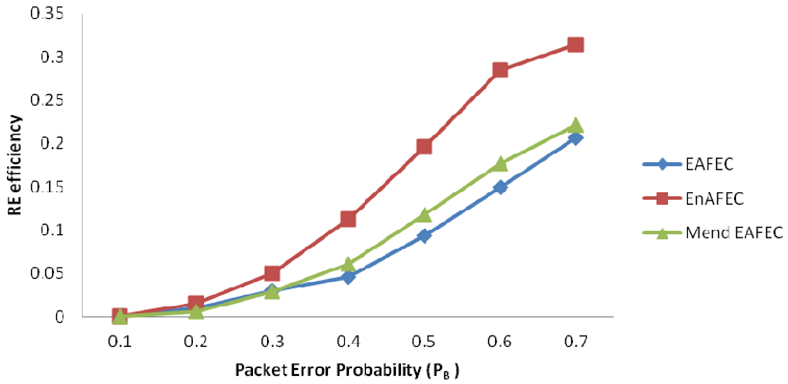

Fig. 9. Recovery Efficiency vs. Error Probability.

FEC efficiency determines how efficiently the FEC packets are used to recover the lost packets. The best value of FEC efficiency is equal to 1. This value indicates full utilization of the FEC packets or refers to a condition whereby there is no FEC packet being transmitted. The condition occurs as the video transmission is free from packet loss. As shown in Table 6 and Fig. 10, the FEC efficiency decreases as the packet error rate increases. This is due to the fact that more video packets are dropped during bad network conditions. It is shown in the figure that EnAFEC achieves the greatest FEC efficiency compared to the others even when the packet error rate is high. When the network condition is good, high smoothing factor value allows lower generation of FEC packets and thereby the number of wasted FEC packets is reduced accordingly. Lower smoothing factor value is required when network condition is bad. The number of FEC packets needs to be increased in order to combat high packet loss. Therefore, EnAFEC mechanism has the ability to adapt with varying wireless network conditions in recovering packet lost. In contrast, Mend FEC contributes the lowest FEC efficiency followed by EAFEC. Therefore, higher FEC efficiency indicates small bandwidth utilization and less congested network because less number of redundant packets is generated.

Table 6. FEC Efficiency.

	EAFEC	EnAFEC	Mend FEC
P_B=0.1	0.9995±0.00015	0.9998±0.000002	0.9885±0.0005
P_B =0.2	0.9972±0.0001	0.9969±0.0001	0.9833±0.0003
P_B =0.3	0.9903±0.0004	0.9889±0.0004	0.9704±0.0005
P_B =0.4	0.9769±0.001	0.9801±0.0006	0.9545±0.0011
P_B =0.5	0.9643±0.0011	0.9757±0.007	0.9426±0.0009
P_B =0.6	0.9539±0.0016	0.9753±0.001	0.9374±0.0006
P_B =0.7	0.9513±0.0017	0.975±0.0008	0.9335±0.0007

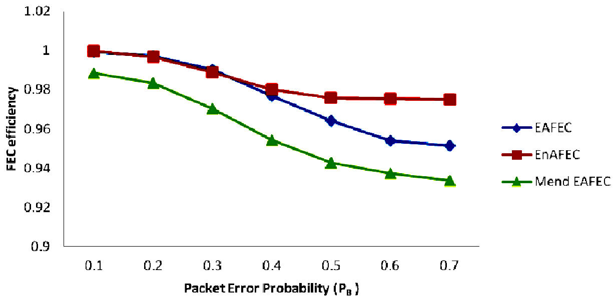

Fig. 10. FEC Efficiency vs. Error Probability.

As shown in Table 7 and Fig. 11, the Mend FEC mechanism achieves high PSNR value compared to the other two mechanisms due to the high number of FEC packets injected into the video transmission. Besides, high FEC packets lead to a high probability of recovering packets from loss. When the P_B is less than 0.3, the EnAFEC achieves the same PSNR value as the Mend FEC. However, when P_B reaches 0.4, the PSNR is reduced due to the reduction of FEC packets in order to avoid network congestion during video transmission. EnAFEC gives better results compared to EAFEC because of its higher error recovery.

Table 7. Peak-Signal-to-Noise Ratio.

	EAFEC	EnAFEC	Mend FEC
P_B=0.1	40.4	40.4	40.4
P_B =0.2	40.2	40.32	40.31
P_B =0.3	39.97	40.06	40.04
P_B =0.4	39.24	39.33	39.46
P_B =0.5	38.32	38.55	38.85
P_B =0.6	36.92	37.1	37.4
P_B =0.7	34.96	35.35	36

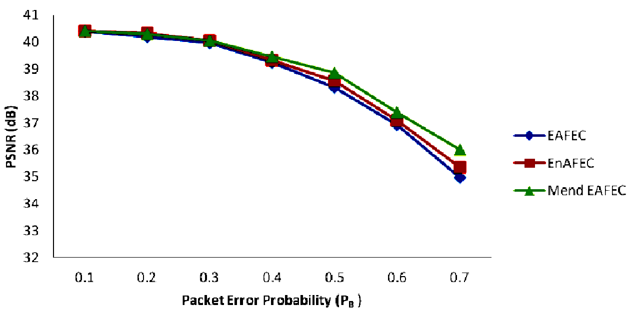

Fig. 11. PSNR vs. Error Probability.

6.1. Comparison: Number of FEC blocks for each video block

Basically, access point generates a certain number of FEC blocks on top of each video packet block. Therefore, it is important for the adaptive FEC mechanism to dynamically adjust the number of redundant packets according to the current network condition in order to avoid network congestion. Fig. 12 plots the number of FEC packets generated at packet error rate of 0.5, which represents a bad network condition.

Based on Fig. 12, EAFEC only generates FEC packets after 40 seconds.That means before 40 seconds, the queue length has reached its threshold setting and no FEC packets will be generated. If packet loss occurs at this time, no error recovery is irrelevant because the unrecoverable loss packets lead to bad video quality at the receiver end. Contrary, the Mend FEC generates FEC packets along the simulation time. At the beginning of the simulation, the FEC blocks generated by the Mend FEC mechanism are low. After a few seconds, the amount of FEC blocks is increased as time increases. For the EAFEC and Mend FEC mechanisms, the maximum number of the generated FEC blocks is equal to 4. Theoretically, only one block of FEC is needed to recover one block of source video. When more than one FEC blocks are injected into the bad network condition, congestion might occur and more video packets are dropped. Thus, the remaining 3 FEC blocks generated by EAFEC is wasted. EnAFEC mechanism has overcome this limitation by generating only one block of FEC packets for each block of video, otherwise none of the FEC packets is generated. This mechanism reduces the amount of FEC blocks to face network congestion. When the queue length reaches the threshold setting, EnAFEC generates a small number of FEC packets. This is because even when the condition buffer at queue length is nearly full, the probability of video packet loss may still occur. Thus, EnAFEC transmits video packet with a small number of FEC packets to reduce the impact of packet loss.

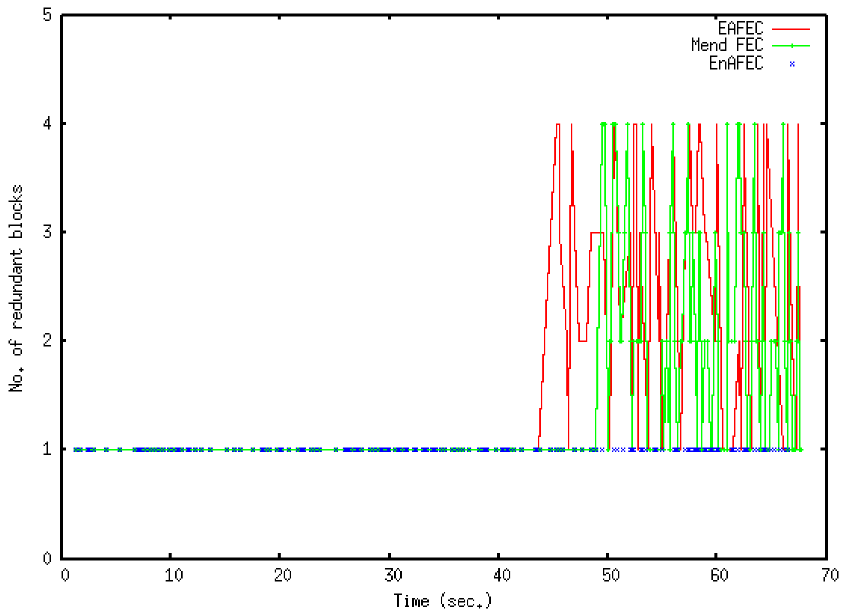

Fig. 12. Number of Redundant Blocks vs. Time.

6.2. Subjective visual quality

Figure 13 illustrates the decoded frame for each FEC mechanism. To compare the effectiveness of all FEC mechanisms, the focus is on the sudden change of the video scene. As the results show, the EnAFEC achieves good video quality with fewer FEC packets to avoid network congestion compared to the EAFEC and Mend FEC. For the EAFEC mechanism, the quality of video produced is bad due to the lost packets which are not recovered. The reason for this situation is that more than one block of FEC packets are generated to recover the missing packet. However, the other video packets are transferred into the network without any error recovery. In fact, too many FEC packets generated for the same original packets yield on low error recovery. On the other hand, the Mend FEC requires more FEC packets to recover the lost packets in order to achieve the most similar video quality produced by the EnAFEC. Overall, EnAFEC has achieved the best video quality with high recovery efficiency.

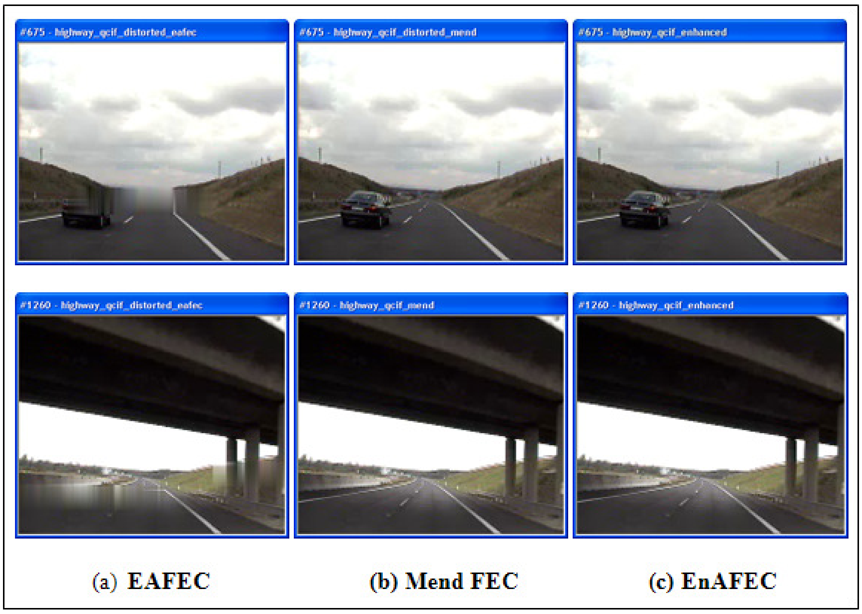

Fig. 13. Comparison in Subjective Visual Quality.

7. Conclusions

After analyzing the results of the experiments, it is found that the performance of the EnAFEC is better than the others in terms of the low FEC packet injected into the network, high FEC efficiency, and high recovery efficiency. The appropriate FEC packets determined by the EnAFEC mechanism reduce the unnecessary extra packets injected into the network when the wireless channel becomes bad. The enhancement is done by implementing block length adaptation and determining appropriate smoothing factor value for queue length. As a result, it shows that EnAFEC mechanism injects the lowest amount of FEC packet into the network for all packet error rate compared to the other EAFEC and Mend FEC. The lowest FEC packet generated by the EnAFEC mechanism provides high recovery efficiency and high FEC efficiency when packet error rate increases, which means the efficiency is nearer to 1. The EnAFEC achieves good video quality with fewer FEC packets in order to avoid network congestion. Apart from that, the performance of EnAFEC, EAFEC and Mend FEC mechanism are evaluated using the GE channel model which is closer to the real wireless error condition that represents the burst wireless network. For the future, the work can be extended by implementing it in real network. The result of the real network experiments may be an interesting issue because it works with real video stream data. On the other hand, this research implements packet error recovery on wireless environment. The packet loss is only considered in a poor channel condition in wireless network, while packet loss that occurs in wired segment is ignored. Usually, packet loss in the wired segment is caused by queue buffer overflow at the router. For the future, this issue would provide the opportunity to study the EAFEC, Mend FEC and EnAFEC in both wired and wireless environment.

References

1. Ding, J.-W.; Chen, W.-J.; and Wang, C.-F. (2006). Adaptive error control for scalable video streaming over wireless internet. *Proceedings of the* 2006 *Joint Conference on Information Sciences, JCIS* 2006, Kaohsiung, Taiwan.
2. Nguyen, T.; and Zakhor, A. (2002). Distributed video streaming with forward error correction. *International Packet Video Workshop* 2002, Pittsburgh, USA.
3. Lin, C.-H.; Ke, C.-H.; Shieh, C.-K.; and Chilamkurti, N.K. (2006). An enhanced adaptive FEC mechanism for video delivery over wireless networks. *International Conference on Networking and Services*, Slicon Valley, California, USA, 106-111.
4. Du, H.; Liu, Y.; Guo, C.; and Liu, Y. (2009). Research on adaptive FEC for video delivery over WLAN. *Proceedings of the* 5*th International Conference on Wireless Communications, Networking and Mobile Computing, WiCom* '09, Beijing, China, 1-4.
5. Latré, S.; Staelens, N.; De Turck, F.; Dhoedt, B.; and Demeester, P. (2007). Improving the quality of multimedia services to wireless users through ahafec deployment. *EuroFGI Workshop on IP QoS and Traffic Control*, Lisbon, Portugal, 135-142.
6. Ge, P.; and McKinley, P. (2002). Comparisons of error control techniques for wireless video multicasting. *Proceedings of the* 21*st IEEE International Performance, Computing, and Communications Conference*, Phoenix, Arizona, 93-102.
7. Han, L.; Park, S.; Kang, S.; and In, H. (2010). An adaptive FEC mechanism using cross-layer approach to enhance quality of video transmission over 802.11 WLANs. *KSII Transactions on Internet and Information System, Korea*, 4(3), 341-357.
8. Subramanian, V.; Kalyanaraman, S.; and Ramakrishnan, K. (2007). Hybrid packet FEC and retransmission-based erasure recovery mechanisms (HARQ) for lossy networks: analysis and design. *International Conference on Communication Systems Software and Middleware*, Bangalore, India, 1-8.

9. Wan, C.-Y.; Eisenman, S.B.; and Campbell, A.T. (2003). CODA: congestion detection and avoidance in sensor networks. *Proceedings* of the 1st *International Conference on Embedded Networked Sensor Systems*, Los Angeles, California, USA, 266-279.
10. Bharghavan. V.; Demers, A.; Shenker, S.; and Zhang, L. (1994). MACAW: a media access protocol for wireless LAN. *Proceedings of the conference on Communications Architectures, Protocols and Applications*, 24(4), 212-225.
11. Karbaschi, G.; and Fladenmuller, A. (2005). Cross layering in wireless multi-hop networks. *International Symposium on Telecommunications* (*1ST* 2005).
12. Jang, H.-C.; and Su, Y.-T. (2008). A hybrid design framework for video streaming in IEEE 802.11e wireless network. *Proceedings* of the 22*nd* *International Conference on Advanced Information Networking and Applications*, Okinawa, Japan, 560-567.
13. Kenguka, K.M.; and Kumchaya, A.S. (2006). Improving WLAN performance with enhanced MAC, node cooperation and two-stage FEC scheme. *Journal of Theoretical and Applied Information Technology*, 2(1), 10-19.
14. Selvakennedy, S. (2006). The influence of MAC buffer on the contention-based access scheme with bursting option for IEEE 802.11E wireless networks. *Journal of Engineering Science and Technology* (*JESTEC*), 1(2), 119-138.
15. Tsai, M.-F.; Chilamkurti, N.; and Shieh, C.-K. (2011). An adaptive packet and block length Forward Error Correction for video streaming over wireless networks. *Wireless Personal Communications*, 56(3), 435-446.
16. Abbas, M.; Chaudhary, N.; Sharma, A.; Venglar, S.; and Engelbrecht, M. (2004). *Methodology for determination of optimal traffic responsive plan selection control parameters.* Texas Transportation Institute, Texas A&M University System.
17. Romdhani, L.; Ni, Q.; and Turletti, T. (2003). Adaptive EDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks. *Wireless Communications and Networking, WCNC* 2003, 2, 1373-1378.
18. Harun, N.Z.; Ghazali, O.; and Osman, B. (2010). Impact of weight values in hybrid and adaptive FEC mechanism over wireless network. *Proceedings* of the *Second International Conference on Network Applications Protocols and Services (NETAPPS)*, 42-47.
19. Ke, C.-H.; and Orozco, J. (2006). A prototype for H.264 evaluation framework using NS2. Retrieved May, from http://140.116.72.80/~smallko /ns2/h264.htm.