An Adaptive FEC Mechanism Using Cross-layer Approach to Enhance Quality of Video Transmission over 802.11 WLANs

Longzhe Han[1], Sungjun Park[1], Seung-Seok Kang[2] and Hoh Peter In[1]
[1] Department of Computer Science and Engineering, Korea University
Seoul 136-713 - ROK
[e-mail: {lzhan, byejune, hoh_in}@korea.ac.kr]
[2] Department of Computer Science, Seoul Women's University
Seoul 139-774 - ROK
[e-mail: msukang@swu.ac.kr]
*Corresponding author: Hoh Peter In

Received March 10, 2010; revised May 6, 2010 and May 28, 2010; accepted June 21, 2010; published December 30, 2010

Abstract

Forward Error Correction (FEC) techniques have been adopted to overcome packet losses and to improve the quality of video delivery. The efficiency of the FEC has been significantly compromised, however, due to the characteristics of the wireless channel such as burst packet loss, channel fluctuation and lack of Quality of Service (QoS) support. We propose herein an Adaptive Cross-layer FEC mechanism (ACFEC) to enhance the quality of video streaming over 802.11 WLANs. Under the conventional approaches, FEC functions are implemented on the application layer, and required feedback information to calculate redundancy rates. Our proposed ACFEC mechanism, however, leverages the functionalities of different network layers. The Automatic Repeat reQuest (ARQ) function on the Media Access Control (MAC) layer can detect packet losses. Through cooperation with the User Datagram Protocol (UDP), the redundancy rates are adaptively controlled based on the packet loss information. The experiment results demonstrate that the ACFEC mechanism is able to adaptively adjust and control the redundancy rates and, thereby, to overcome both of temporary and persistent channel fluctuations. Consequently, the proposed mechanism, under various network conditions, performs better in recovery than the conventional methods, while generating a much less volume of redundant traffic.

Keywords: Forward error correction, cross-layer, video streaming, wireless networks

This study was jointly funded by the R&BD Support Center of Seoul Development Institute and the South Korean government (Project title: WR080951, Establishment of Bell Labs in Seoul/Research of Broadband Convergent Networks and their Enabling Technologies), and by the Korea Research Foundation Grant, which in turn is funded by the Korean Government (MOEHRD; KRF-2007-211-D00099 and KRF-2007-331-D00360 (Basic Research Promotion Fund)). A preliminary version of this paper appeared in ICONI/APIC-IST, 209-215, Dec 2009.

DOI: 10.3837/tiis.2010.06.009

1. Introduction

As technology advances concerning wireless networks and mobile computing devices, multimedia services over Wireless Local Area Networks (WLANs) have gained worldwide popularity. Specifically, factors such as the increasing bandwidth, low cost and flexibility have propelled wide deployment of the IEEE 802.11 WLAN for connecting wireless users to the Internet. In the meanwhile, Internet multimedia services are in high demand to support diverse wireless network environments, such as Video-on-Demand (VoD), Internet Protocol Television (IPTV), video game and video conferencing.

However, these services have to overcome diverse challenges in design, which are imposed by, among others, burst packet losses, channel fluctuation and limited Quality of Service (QoS) support on wireless networks. In addition, what matters more complicated are the bandwidth-intense, loss-tolerant and delay-sensitive characteristics of the video streaming applications [1]. Especially, the high rate of packet loss significantly decreases the network throughput and video quality, a phenomenon that is caused by signal fading, hidden terminal and interference. Two basic approaches (i.e. packet-level Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ)) are commonly used to cope with packet errors on wireless networks [2][3]. The ARQ mechanism is conceived to transfer data in a reliable way. The approach offers two ways of activating a retransmission: upon request from the receiver, and upon timeout of the timer at the sender. The delay arising out of retransmission renders the ARQ mechanism inappropriate for highly delay-sensitive video streaming applications. On the contrary, the FEC approach obviates retransmission by means of sending, to the receiver, additional error-correcting information, along with the original data. Thus, it becomes possible for the receiver to recover a certain number of lost packets from the received ones. Due to the loss-tolerant characteristic of video streaming applications, the FEC is more suitable than the ARQ for the video transmission over wireless networks [4].

The FEC scheme deals with the issue of packet loss by utilizing redundant data. Conveying of these redundant FEC packets consumes additional network resources such as network bandwidth, packet-buffer memory and wireless device power. As wireless channel fluctuations (i.e. fast fading and slow fading) lead to frequent occurrence of errors and burst packet losses, the two phenomena severely hamper the efficiency of an FEC mechanism. Traditionally, the FEC function is implemented on the application layer. The redundancy rate is either static or controlled by application-layer programs, based on feedback information such as acknowledgements from the transport layer or the application layer. The delay in transmitting acknowledgement makes it difficult to make adjustments to current network conditions. The conventional FEC approaches, however, may not effectively recover lost packets, because they fail to capture the real-time network conditions and to adjust the redundancy rates accordingly. It is crucial for an effective and efficient FEC mechanism to be able to accurately detect channel fluctuations and to dynamically manipulate the redundancy rates.

To accomplish the goal, this study introduces an Adaptive Cross-layer FEC mechanism (ACFEC), which is to enhance the quality of video streaming over the IEEE 802.11 WLANs. The proposed ACFEC mechanism is implemented at the wireless Access Point (AP). Since data packets run through the wireless AP in the infrastructure mode, it is possible to monitor traffic flows and to capture ever-changing channel conditions with the wireless AP. By employing a cross-layer scheme, our proposed ACFEC mechanism leverages the functionalities of different layers. Because the ARQ function on the MAC layer is able to effectively detect lost packets, the ACFEC mechanism retrieves the information on the loss from the MAC layer, and adaptively controls the redundancy rates in accordance with the current network conditions. In contrast to conventional approaches, the ACFEC mechanism would not generate any unnecessary FEC packets, unless errors occur during video transmission.

The rest of the paper is organized as follows: Section 2 discusses related work on the forward error correction strategies and cross-layer approaches; the proposed ACFEC mechanism is introduced in Section 3; Section 4 presents the simulation settings and analyzes experimental results; and, finally, conclusions and future work are discussed in Section 5.

2. Related Work

2.1 Forward Error Correction

The FEC scheme has been adopted to address issues of packet error and loss, and to improve the quality of video delivery. The goal is achieved by adding error-correcting information to the original data. The FEC is usually performed at the byte level and at the packet level [2] [5]. The byte-level FEC recovers bit errors in one packet, and functions on the link layer of the WLANs. This study focuses on the packet-level FEC mechanism, which mainly deals with packet loss. **Fig. 1** outlines the conceptual structure of the FEC function. The source data stream is divided into separate blocks, with each block consisting of *k* source packets.

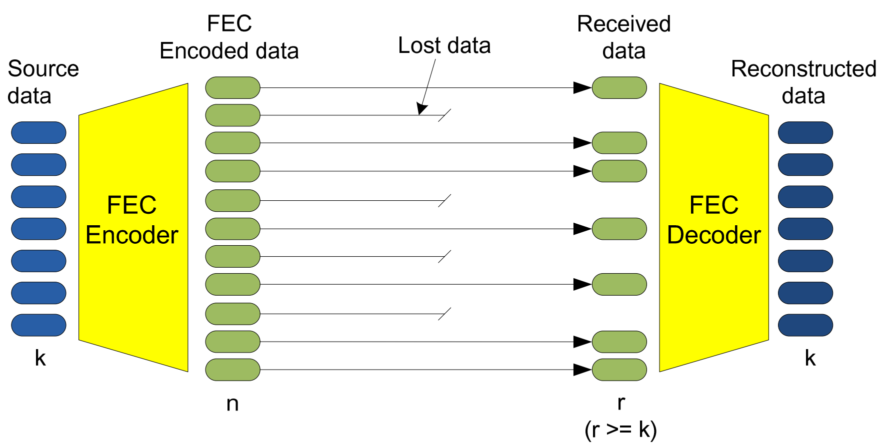

Fig. 1. Pack-level FEC Process

For every set of *k* source packets, the FEC encoder generates *n* redundant packets. During transmission of one block consisting of *n* packets, the FEC decoder is able to regenerate the source data, unless the number of lost packets in this block is greater than the quantity of (*n* – *k*). Widely in use for the packet-level FEC are block-based error correction codes such as simple parity check and Reed-Solomon [6]. Let's put packet-level FEC strategies into two broad categories: static approach and adaptive approach. In the case of the static FEC, the redundant rate is fixed and the FEC encoder always generates the same amount of FEC packets. Its advantage lies in its easiness to implement. A trade-off follows between the resistibility to packet loss and the efficient utilization of network resources. More lost packets can be tolerated by increasing the quantity of (n – k). However, it consumes more resources to transmit a larger number of redundant packets, such as network bandwidth, packet-buffer memory and wireless device power. Moreover, if congestion arises over the wireless network, an attempt to transmit those redundant packets might end in the decreased efficiency of the FEC and in worsened congestion. To address this issue, many studies have proposed solutions built around the notion of the adaptive FEC [7][8][9][10][11][12]. K. Park et al. [7] proposed an adaptive FEC mechanism to achieve the QoS support for end-to-end transport of MPEG video (MAFEC). The MAFEC dynamically controls the FEC rate in accordance with the feedback information, which causes a real-time delay in transmission. W. Kumwilaisak et al. [8] designed an adaptive protection scheme, based on channel condition estimation. By feeding back the information on signal strength, it becomes possible to capture the channel conditions, such as short-term fading, long-term fading and packet loss. In daily normal environments, the end users may have various distances from the video source. The transmission delay of feedback information makes it difficult to reflect current network conditions. Because the network conditions are constantly changing in a wireless network environment, the conventional FEC approaches, therefore, may not function effectively. In [9], the authors proposed a novel adaptive FEC mechanism titled "RED-FEC." The RED-FEC is conceptually constructed around the idea of random early detection algorithm [13], which uses the queue length of the wireless AP as the congestion indicator. Under this scheme, the redundancy rate is computed, using the queue length. The RED-FEC gradually reduces the redundancy rates, as the queue length increases.

2.2 Cross-Layer Approach

On the constant rise is the number of applications running on the wireless Internet. Meanwhile, different applications have various QoS requirements. For example, video streaming (e.g. VoD and IPTV) requires supports for minimum bandwidth and end-to-end delay guarantee, while reliable data transmission has high priority in web browsing, messaging and file transfer. In addition, as illustrated by the simulation results in Section 4.4, the original TCP may not work properly on wireless networks. This is because the original TCP/IP protocol suite assumes a layered architecture. Each layer provides services to other layers through specifically defined interfaces, such as networking application programming interface (e.g. socket programming). Although the layered architecture approach offers diverse merits in terms of system design and software development, the information encapsulated in each layer may cause problems over a wireless network. The congestion control algorithms of the TCP include slow-start, congestion avoidance, fast retransmission and fast recovery. By measuring end-to-end delays and packet drops, the algorithms perform exceedingly well on wired networks. In a wireless network environment, however, the TCP processes packet drops caused by congestion as it does with packet losses caused by wireless transmission errors. In the presence of high rates of errors, the TCP frequently enters into the congestion avoidance procedure, and, thereby, reduces data transmission rates. The unnecessary data rate reduction significantly affects throughput and bandwidth utilization [14]. Cross-layer approaches have been proposed to provide various QoS supports and to overcome the limitations posed by wireless networks [15][16][17]. By leveraging the functionalities of different network layers, cross-layer approaches have achieved significant performance-wise improvements. In [15], the authors offer an excellent analysis of the existing approaches and their solutions for multimedia data transmission over wireless networks. The authors further classify cross-layer architectures into the following five categories:

- Top-down approach: Based on the QoS requirements, a higher layer protocol selects the parameters and strategies of the next lower layer.
- Bottom-up approach: Under this approach, a lower layer performs optimization without triggering awareness on the higher layers.
- Application-centric approach: Tailored for the characteristics of an application, the corresponding application layer protocol optimizes parameters and strategies of lower layers.
- MAC-centric approach: In this scheme, the MAC layer determines the QoS level of each transmission flow, based on the requirements of the application layer.
- Integrated approach: All network layers involved determine the parameters and strategies jointly.

3. Adaptive Cross-Layer FEC Mechanism

3.1 Proposed Cross-Layer Architecture

Unlike the conventional FEC approaches in which FEC functions are implemented on the application layer, our proposed ACFEC mechanism operates at the wireless AP, as also used in [9]. As defined in [15], the ACFEC architecture follows therein the paradigm of the integrated approach, and focuses on utilizing functionalities of different layers. **Fig. 2** shows that the adaptive FEC controller cooperates with the User Datagram Protocol (UDP) and the Medium Access Control (MAC) protocol. The cooperation is achieved through the mechanism proposed in [17]. The adaptive FEC controller consists of two components: FEC packet generator and packet loss monitor.

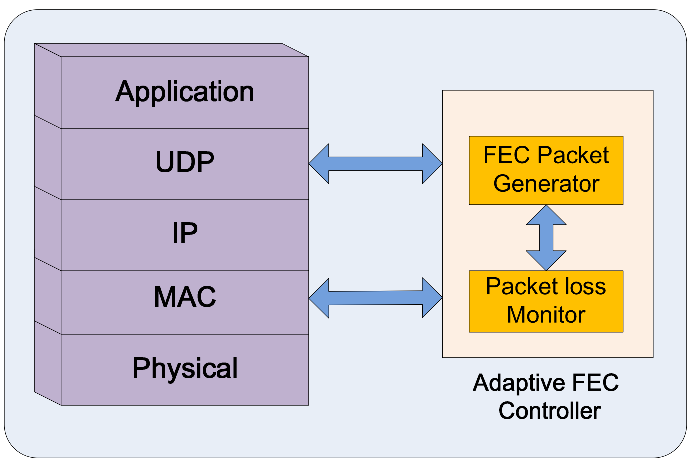

Fig. 2. Adaptive Cross-Layer FEC Architecture

The streaming server encapsulates the video data in Real-time Transport Protocol (RTP) packets, and delivers them to the receiver through the wireless AP. When a packet arrives at the AP, the adaptive FEC controller retrieves the packet header from the UDP, and identifies the packet type by checking the RTP header. The standard IEEE 802.11 [18] defines MAC-Level acknowledgments in the unicast transmission mode to recover errors, and the acknowledgements are a stop-and-wait ARQ mechanism in nature. Once a frame is sent out, the AP does not send any further frames until it receives an acknowledgement (ACK) from the receiver. If the AP fails to receive the ACK prior to timeout, the same frame is sent again.

Then, the ARQ process stops, if the retransmission counter reaches the RetryLimit [18], and the frame is regarded as lost. However, traditional transport protocols (TCP and UDP) have not utilized this information. The adaptive FEC controller retrieves the failure information, and uses it to adaptively control the redundancy rates.

3.2 Adaptive FEC Mechanism

The packet-level FEC encoder generates error correction packets, based on a certain number of source packets that constitute a single block. **Fig. 1** depicts the idea of the basic FEC mechanism. Because the ARQ function of the MAC layer is able to detect packet losses, it is possible to measure the average channel loss rate P_{ch}. The average packet loss rate is given by:

$$P_{pkt} = (P_{ch})^r \tag{1}$$

where r refers to the RetryLimit. The probability of successful transmission of a packet to the receiver is given by $(1 - P_{pkt})$. m is assumed to represent the number of lost packets in one block. The following expression illustrates the relation of redundant FEC packets h for recovery of m lost packets:

$$(1 - \mathrm{P_{pkt}}) \times h = m \tag{2}$$

From (1) and (2), the amount of redundant FEC packets required for recovering m lost packets is given by:

$$h = \frac{m}{\left(1 - (P_{ch})^r\right)} \tag{3}$$

The detailed process of our proposed ACFEC mechanism is presented in **Fig. 3**. When source packets are conveyed to a receiver through a wireless AP, the adaptive FEC controller classifies video data packets, and groups them in blocks. Then, the video data packets of an ongoing block are stored in the buffer for generation of FEC packets. The packet loss monitor traces the transmission results of the video data packets by snatching up the failure information from the MAC layer. If a packet is lost, the failure counter of the packet loss monitor increases by one. The failure is caused by the ARQ mechanism of the MAC layer is unable to recover the transmission error and reach to the RetryLimit. After conveying one block of the video data packets, the packet loss monitor calculates, through Equation (3), the number of redundant FEC packets to be generated. Because the packet loss monitor detects the actual number of packet losses in each block, the ACFEC mechanism becomes capable of effectively producing relevant amounts of FEC packets for various quantities of packet loss. In order to understand the exact volume of conveyed redundant data, we have investigated how many FEC packets should be generated through simulation experiments shown in section 4.2.2.

If the current queue length has not reached the threshold, the FEC packet generator produces the required FEC packets, and transmits them to the receiver. By accurately detecting packet losses and adjusting the redundancy rates accordingly, the number of FEC packets is increased or decreased to meet the actual need of the receiver, and to compensate for the packet losses. No FEC packets are produced, when all of the video data packets of one block arrive at the receiver successfully. As discussed in the next section through the experimental results, the ACFEC mechanism, under various network conditions, performs better in recovery than the conventional methods, while generating a much smaller volume of redundant FEC traffic.

```
When a packet arrive:

 If (type(packet) == VIDEO) then
   SaveToBuffer(packet)
   Send(packet)
   If (transmission failed) then
     failure_counter = failure_counter + 1
   End If
   If (a block is transmitted) then
     If (failure_counter > 0) then
       If (queue_length < threshold) then
         num_FEC = CalculateFEC(failure_counter)
       Else
         num_FEC = 0
       End if
       Send_FEC_Packet(num_FEC)
     End If
     Clear(buffer)
     failure_counter = 0
   End If
 End If
```

Fig. 3. Pseudo-Code for ACFEC

3.3 Queue Management Policy

The queue management policy adopted by the wireless AP will influence the network throughput, end-to-end delay and QoS. At the wireless AP, the scheduling algorithm decides and selects the next packet to be transmitted. It also controls how to drop the packets when the queue space is filled up. There are many scheduling policies over a wired network such as First In First Out (FIFO), Fair Queueing (FQ), Stochastic Fairness Queueing (FQ) and Class-Based Queueing (CBQ). On the contrary, only FIFO and priority queue are used on WLANs. The IEEE 802.11e standard provides QoS supports by defining four access categories with different transmission priorities [19]. In this paper, we mainly focus on widely deployed IEEE 802.11b WLANs alone, and use the priority queue in the wireless AP. The priority queue implemented in the NS-2 simulator [20], PriQueue, has only one queue. The PriQueue gives the routing protocol packets a higher priority, and inserts the packets into the head of the queue. Lower priority is assigned to other packets which employed the FIFO policy.

The design of the proposed ACFEC mechanism considers both wireless errors and network congestion. Since the PriQueue simply drops the packets at the tail when the queue space is full, the ACFEC mechanism needs to control the generation of FEC packets. If the network is overloaded, injecting many FEC packets into the network may not improve the error recovery rates. Instead, it results in the decreased video quality. Because the queue length is an effective indicator of network congestion [13], the ACFEC mechanism uses the threshold that is proportional to the queue length to control the FEC packets, as shown in **Fig. 3**. If the current queue length is larger than the threshold, no FEC packets are delivered. As the queue space is almost filled up, the FEC packets have high chances being dropped by the

PriQueue. The transmitted FEC packets may not suffice to recover the packet losses. To improve the bandwidth utilization and the network throughput, the ACFEC will not generate any FEC packets when the network is congested. In order to obtain the appropriate threshold value, we conducted extensive experiments. The simulation settings and results are described in the next section.

4. Simulation and Results

The NS-2 simulator [20] was employed to evaluate the proposed ACFEC mechanism, and to compare it both with the static FEC mechanism and with the RED-FEC mechanism. The static FEC mechanism generates a fixed number of redundant packets for each source data block. In order to correctly simulate the RED-FEC mechanism in the NS-2, we have referred to [21] and have adopted the parameter settings set forth in [9].

4.1 Experimental Environment

The “Foreman” CIF and “Highway” QCIF sequences are used as the video traces [22], and encoded, using H.264 standard. Each frame encapsulated in the RTP/UDP/IP packet. The frame rate of the encoded sequences was 30 frames/sec. The video server transmits video to the receiver via unicast delivery over infrastructure-based WLANs by means of the Distributed Coordination Function (DCF). For the light background traffic, there was one CBR flow over the network. On the other hand, for the heavy background traffic, one FTP flow and two CBR flows were injected. The link between the AP and the receiver was IEEE 802.11b 11Mbps. The cap on retransmission (RetryLimit) is set to 4 in the simulation, which represents the default number under the IEEE 802.11 standard [18]. The detailed settings are shown in **Table 1**.

Table 1. The NS2 simulation environment

Parameter	**Setting**
Antenna type	OmniAntenna
Propagation model	TwoRayGround
Network size	500m X 500m
MAC protocol	802.11b without RTS/CTS
RetryLimit	4
MAC bandwidth	11 Mbit
IFQ length	50
Interface queue type	Drop-tail priority
Routing protocol	NOAH
Video trace	“Foreman” CIF with 300frames “Highway” QCIF with 2000 frames

4.1.1 Wireless Error Model

We have adopted the Gilbert-Elliott model in our experiments to simulate wireless errors. The packet loss pattern on WLANs exhibits the burst property. For example, if a previous packet is lost, then the probability of losing the following packet rises as well. In order to capture this attribute, [23] proposed a two-state Markov model, also known as the Gilbert-Elliott model.

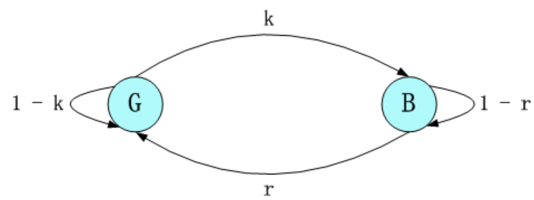

Fig. 4. Two-State Markov Model

As shown in **Fig. 4**, the state *G* means that a packet has arrived correctly, while the state *B* indicates that a packet has been lost. *k* denotes the probability of the state transition from *G* to *B*, while *r* means the opposite. The steady state probabilities for the states *G* and *B* are computed, as follows:

$$\pi_G = \frac{r}{r+k} \quad \text{and} \quad \pi_B = \frac{k}{r+k} \tag{4}$$

The average packet loss probability is:

$$P_{avg} = P_G \pi_G + P_B \pi_B \tag{5}$$

where P_G and P_B indicate the probabilities of lost packets in the states *G* and *B*, respectively.

4.2 Analysis of Simulation Results

4.2.1 Comparison: Aggregate Totals of FEC Packets

For performance experiments of the ACFEC with other approaches, the “Foreman” CIF sequence was used. **Fig. 5** and **Fig. 6** depict the number of redundant FEC packets transmitted both under a light and a heavy traffic load with different packet loss rates. When the traffic load is light, as shown in **Fig. 5**, the static FEC and the RED-FEC mechanisms produce similar results. The ACFEC gradually increases the number of FEC packets, as the loss rate increases. In addition, there will be no FEC packets generated in the absence of packet losses. It follows that our proposed mechanism is able to capture the channel fluctuation, and adaptively controls the redundancy rate.

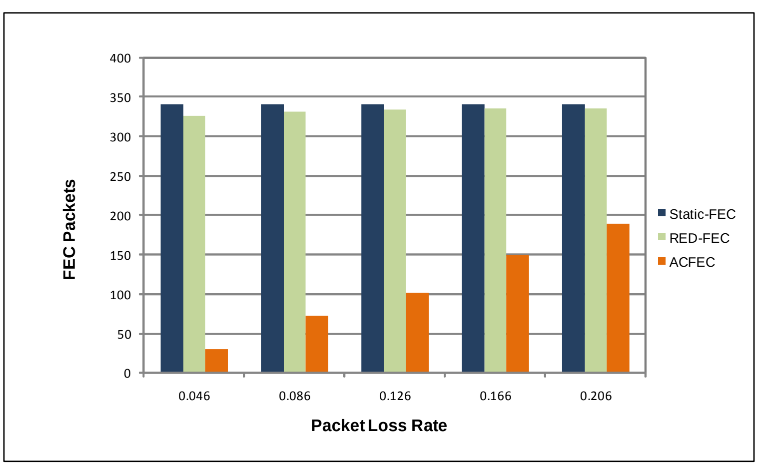

Fig. 5. FEC Packets under Light Load

Under a heavy traffic load, the static FEC still generates a fixed number of FEC packets all the time. Thus generated, the FEC packets, together with the background traffics, cause congestion frequently occurring at the AP. Therefore, the AP discards some of the FEC packets. **Fig. 6** demonstrates this process. In case that the packet loss rate is 0.046, the number of the FEC packets transmitted through the static FEC remains below the one under light traffic. However, as explained in Section 2.2, the original TCP flow is to slow down dramatically in the presence of high error rates. In **Fig. 6**, for the static-FEC, the number of FEC packets increase, as packet loss rate rises. This reaction occurs, because the transmission speed of FTP flow gets reduced upon increase of the loss rate so that more FEC packets can go through the AP. On the contrary, both of the ACFEC and the RED-FEC function to avoid congestion. When the packet loss rate is low, however, the RED-FEC mechanism conveys a much smaller amount of FEC packets, a sharp contrast that does not appear under a light load of traffic. It follows, therefore, that the RED-FEC increases the number of FEC packets, as the queue length of the AP reduces.

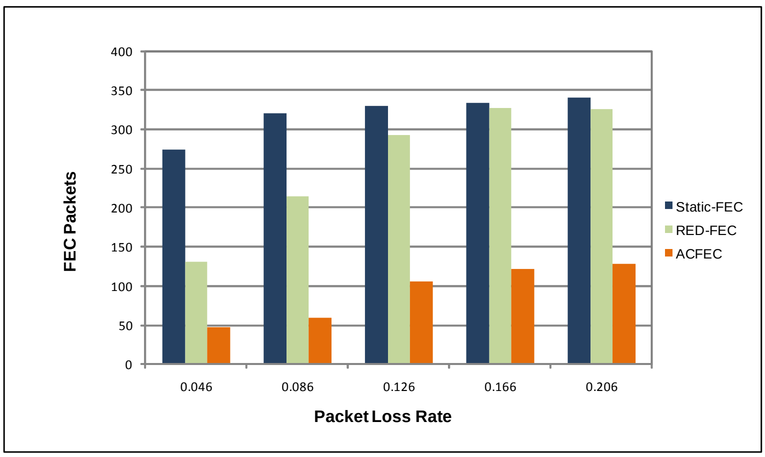

Fig. 6. FEC Packets under Heavy Load

4.2.2 Comparison: Number of FEC Packets for Each Block

As described in Section 2.1, the FEC encoder produces a certain number of redundant FEC packets for each block of source packets. It is important for the adaptive FEC mechanisms to dynamically adjust the redundancy rates in compliance with current network conditions. The results described in this section are acquired, when the traffic load is heavy.

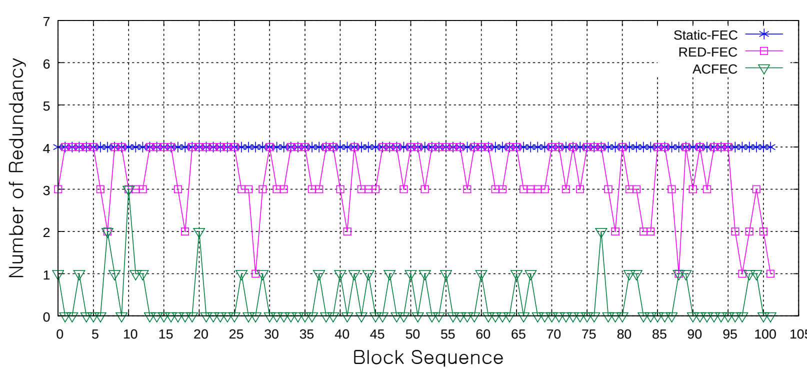

Fig. 7. Number of Redundant Packets at Loss Rate of 0.046

Fig. 7 plots the amounts of FEC packets generated under each approach, with the packet loss rate at 0.046. In this case, few source packets are lost in one block; accordingly the ACFEC produces a small quantity of FEC packets. If all source packets are successfully delivered, the ACFEC skips the FEC encoding process. The RED-FEC reduces the redundancy rates, faced with congestion on the network. However, the Static-FEC always transmits a fixed amount of FEC packets.

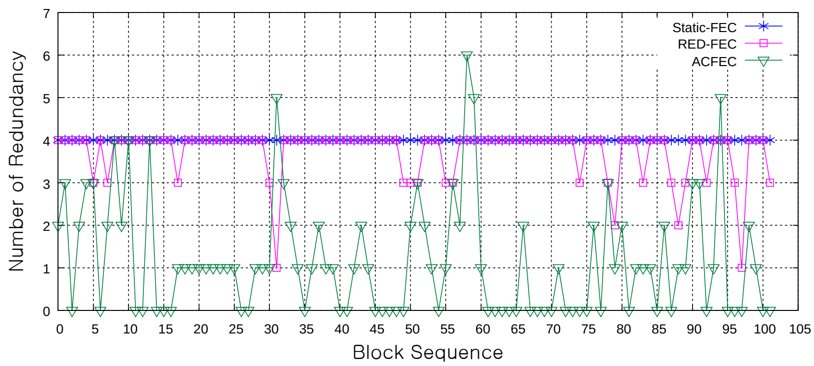

Fig. 8. Number of Redundant Packets at Loss Rate of 0.126

When the packet loss rate rises, as shown in **Fig. 8**, the ACFEC adaptively increases the number of FEC packets to compensate for the lost packets in each block. **Fig. 9** illustrates that the source packets suffer from a severe burst loss. Adjacent packets belonging to the affected block might be lost together. In response, the ACFEC effectively detects the burst packet losses, and adjusts the redundancy rates accordingly. Under the same circumstances, the limited redundant FEC packets, which have been produced by the Static-FEC and RED-FEC, do not suffice to compensate for the lost packets.

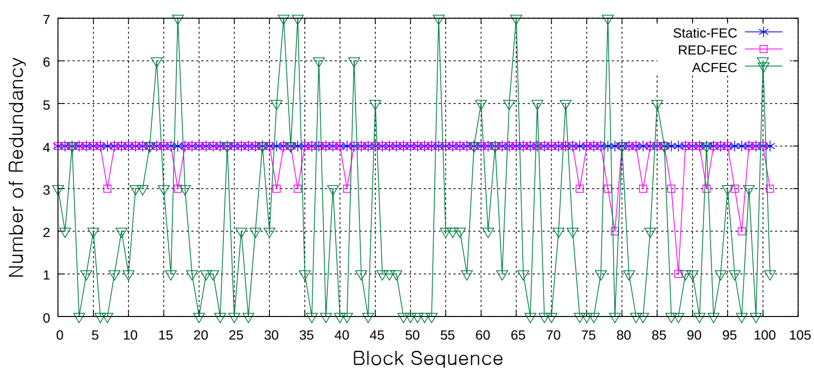

Fig. 9. Number of Redundant Packets at Loss Rate of 0.206

4.2.3 Analysis of FEC Efficiency

In order to measure the impact that the redundant FEC packets carry in the course of overcoming the packet losses, we have adopted the factor of FEC efficiency, which is defined in [4], as follows:

$$FEC_{effective} = \frac{SourcePacket_{total}}{SourcePacket_{received} + FECPacket_{transmitted}} \tag{6}$$

where $SourcePacket_{total}$ indicates the amount of packets delivered to the application layer programs. That includes the successfully received packets ($SourcePacket_{received}$) and the packets recovered by FEC packets. $FECPacket_{transmitted}$ refers to the total number of the FEC packets transmitted.

Fig. 10 illustrates the efficiency of each FEC mechanism under different rates of packet loss. The results are acquired, when the traffic load is heavy. The RED-FEC performs more efficiently than the Static-FEC does, when the loss rate is low. The $FEC_{effective}$ of the ACFEC stands at 0.97 at a loss rate of 0.046, and it gradually slides down to 0.79, as packet losses rise.

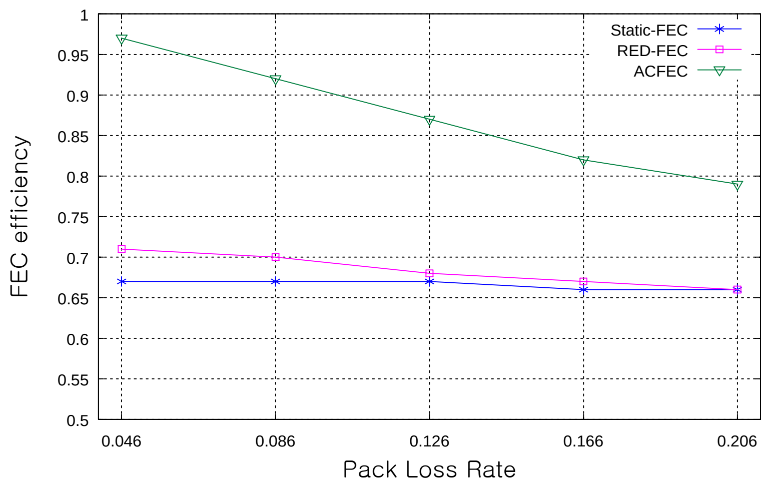

Fig. 10. FEC Efficiency under Heavy Load

4.2.4 Comparison of the PSNR

The Peak Signal-to-Noise Ratio (PSNR) is used as a criterion for measurement of video quality. **Fig. 11** depicts the PSNR values under light traffic. The Static-FEC, the RED-FEC and the proposed ACFEC are all capable of recovering small quantities of lost packets at a loss rate of 0.046. Therefore, all of the three approaches produce the same values. However, the PSNR values of the Static-FEC and the RED-FEC decrease faster than the ACFEC does, as the loss rates rise. The average PSNR of the ACFEC stands at 35.57 dB. For the Static-FEC and the RED-FEC, the average PSNR values are 32.74 dB and 31.93 dB, respectively.

Fig. 12 presents the proposed ACFEC also obtained better performance than the existing approaches in the heavy load condition. The ACFEC gradually decreases the video quality as the packet loss rates increase. The result of the RED-FEC is similar to the static-FEC, while the REC-FEC is capable of adjusting the redundant rates, based on the volume of a traffic load. The average of PSNR values for each approach is shown in **Table 2**. Finally, **Fig. 13** and **Fig. 14** represent decoded frames of each approach.

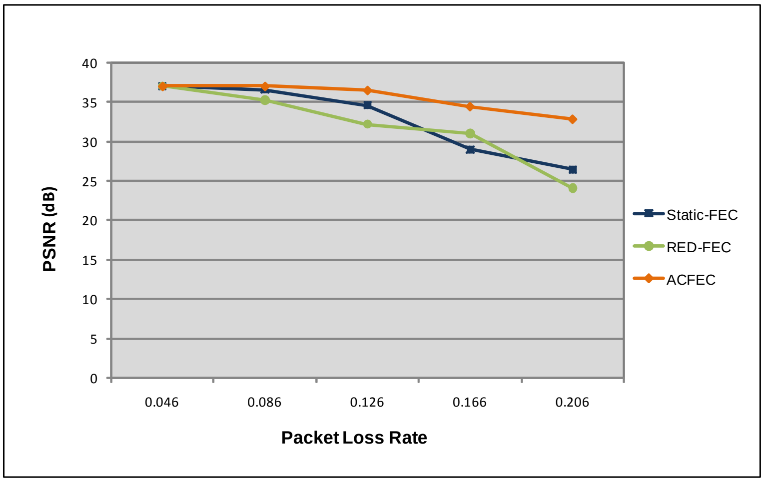

Fig. 11. PSNR under Light Load

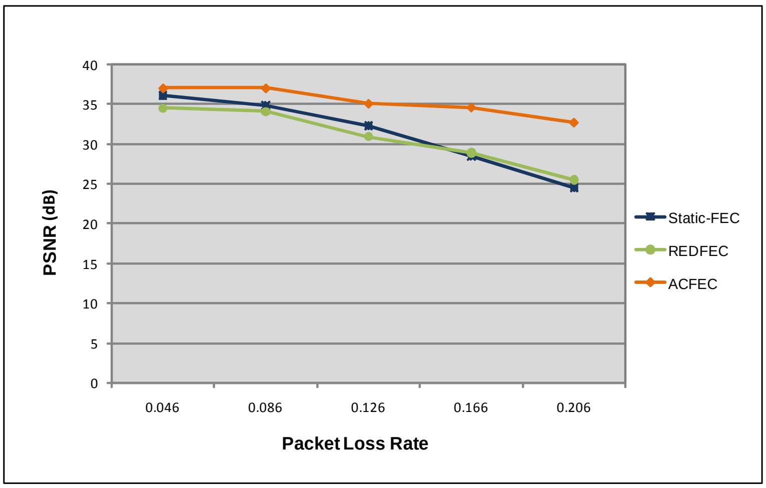

Fig. 12. PSNR under Heavy Load

Table 2. Average PSNR Values

Approach	Light load (dB)	Heavy load (dB)
ACFEC	35.57	35.29
RED-FEC	31.93	31.23
Static-FEC	32.74	30.78

(a) Static-FEC (b) RED-FEC (c) ACFEC

Fig. 13. Light Load at a 0.206 Loss Rate

(a) Static-FEC (b) RED-FEC (c) ACFEC

Fig. 14. Heavy Load at a 0.206 Loss Rate

4.3 Queue Threshold Analysis

In order to choose the appropriate threshold as discussed in section 3.3, we carried out the experiments with the heavy-load condition and a 20% packet loss rate. The video sequence was "Highway" in the QCIF format with 2000 frames [22].

Four threshold values (0.7, 0.8, 0.9 and 1.0) were compared in the experiments. "0.7" means that, if the packets fill 70% of the queue, no more FEC packets are generated. "1.0" implies that it always produces and delivers the amount of FEC packets which is calculated by Equation (3). **Fig. 15** shows the impact of the different threshold values to the queue length of the AP. As pointed out by [13], the average queue length is sensitive to the characteristics of network traffics (such as TCP/UDP flows and various transmission rates). The threshold 1.0 has no effect of the FEC generation. Therefore the average queue length of the threshold 1.0 is larger than others. By using the threshold values 0.7, 0.8, and 0.9, the ACFEC can control the redundant rate, when the AP is congested. However, **Fig. 16** presents the video quality achieved on different threshold values. Clearly, the most appropriate value is "0.9", which was used in our experiments.

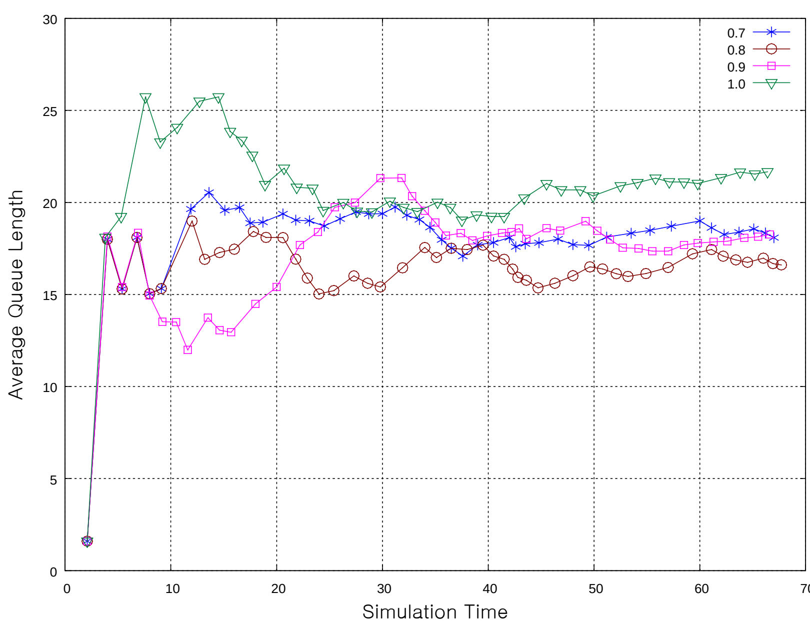

Fig. 15. Average Queue Length on Different Threshold Values

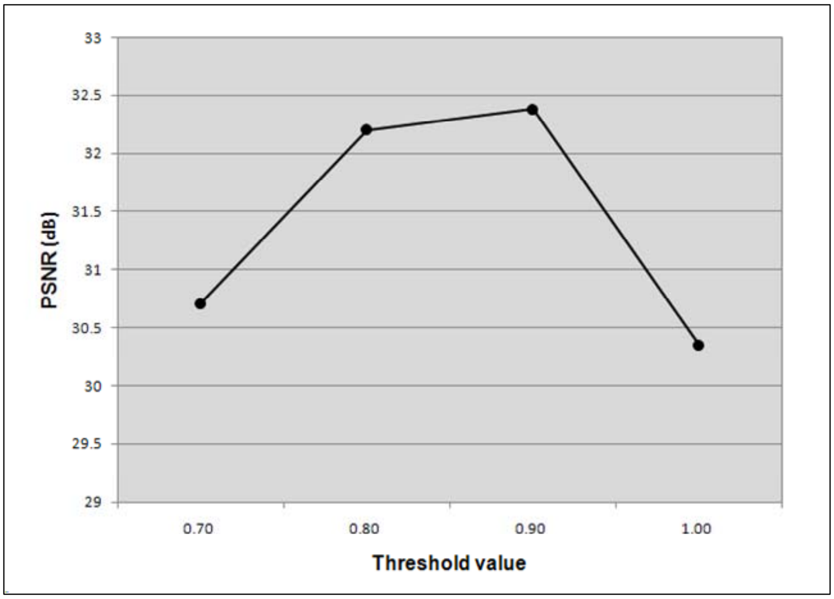

Fig. 16. PSNR on Different Threshold Values

5. Conclusion

In this study, we have proposed an Adaptive Cross-Layer FEC mechanism (ACFEC) to enhance the quality of video transmission over 802.11 Wireless Local Area Networks (WLANs). Our proposed ACFEC mechanism utilizes the functionalities of different layers. The Automatic Repeat reQuest (ARQ) function is applied on the MAC layer to detect lost packets. Through cooperation with the UDP protocol, the redundancy rates are adaptively controlled based on the loss information. The experimental results demonstrate that our ACFEC mechanism adaptively controls the redundancy rates to overcome channel fluctuations, while producing a much smaller volume of redundant traffic.

In the future, we plan to extend the ACFEC mechanism onto the H.264 Scalable Video Codec (SVC), because the scalabilities of the SVC will enhance the quality of video streaming over WLANs. We also consider studying how to utilize the QoS function supported by IEEE 802.11e under the ACFEC mechanism.

References

[1] Qian Zhang, Wenwu Zhu, Ya-Qin Zhang, "End-to-End QoS for Video Delivery Over Wireless Internet," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 123-134, 2005.
[2] C. Huitema, "The Case for Packet Level FEC," in *Proc. of the 5th Workshop in Protocols for High-Speed Networks*, pp. 109-120, 1996.
[3] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ Error Control for Fading Mobile Radio Channels," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 445-455, 1997.
[4] A. Nafaa, T. Taleb, and L. Murphy, "Forward Error Correction strategies for Media Streaming over Wireless Networks," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 72-79, 2008.
[5] A. Basalamah and Takuro Sato, "A Comparison of Packet-Level and Byte-Level Reliable FEC Multicast Protocols for WLANs," in *Proc. IEEE Globecom'07*, pp.4702-4707, Nov. 2007.
[6] Wicker and Bhargava, "Reed-Solomon Codes and Their Applications," *IEEE Press*, 1994.
[7] Kihong Park, Wei Wang, "QoS-Sensitive Transport of Real-Time MPEG Video using Adaptive

Forward Error Correction," in *Proc. of the IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)* , vol. 2, pp. 426, 1999.

[8] W. Kumwilaisak, J. Kim, and C.-C. Jay Kuo "Reliable wireless video transmission via fading channel estimation and adaptation," in *Proc. of WCNC 2000*, Chicago, USA, Sep. 2000.

[9] C. H. Lin, C. K. Shieh, N. Chilamkurti, C. H. Ke, W. S. Hwang, "A RED-FEC Mechanism for Video Transmission over WLANS," *IEEE Transaction on Broadcasting: Quality Issues in Multimedia Broadcasting*, vol. 54, no. 3, pp.517-524, 2008.

[10] S. Fosse-Parisis, J.-C. Bolot and D. Towsley, "Adaptive FEC-based error control for Internet telephony," *In Proc. IEEE INFOCOM*, vol. 3, pp. 1453-1460, 1999

[11] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, pp. 1707-1723, October 1999.

[12] Myoung-Kyoung Ji, Tae-Uk Choi, So-Huyn Lee, Seong-Ho Park, Ki-Dong Chung, "Adaptive FEC Error Control Algorithm Using Loss and Delay," in *Proc. of the IEEE International Conference on Multimedia and Expo (ICME'01)*, pp.22, 2001.

[13] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 4, pp. 397-413, Aug. 1993.

[14] Hari Balakrishnan , Srinivasan Seshan , Elan Amir , Randy H. Katz, "Improving TCP/IP performance over wireless networks," in *Proc. of the 1st annual international conference on Mobile computing and networking,* p.2-11, Berkeley, California, USA, Nov. 13-15, 1995.

[15] M. Van Der Schaar et al., "Cross-Layer Wireless Multimedia Transmission: Challenges, Principles, And Paradigms." *IEEE Wireless Commun.*, Aug. 2005.

[16] Y. Shan and A. Zakhor, "Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks," in *Proc. IEEE ICME 2002*, Aug. 2002.

[17] S. Shakkottai, T. S. Rappaport and P. C. Karlsson, "Cross-Layer Design for Wireless Networks," *IEEE Commun. Mag.*, Oct. 2003.

[18] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007*, June 2007.

[19] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements," *IEEE Std 802.11e-2005*,November 2005

[20] The Network Simulator 2 ns-2, http://www.isi.edu/nsnam/ns/

[21] NS-2 Adaptive FEC Tool, Online: http://140.116.72.80/~jhlin5/ns2/EAFEC/fec_simulation.htm

[22] YUV Video Sequences, Online: http://trace.eas.asu.edu/ yuv/index.html

[23] H. Sanneck and G. Carle, "A Framework Model for Packet Loss Metrics based on Loss Run Length," Proc. *SPIE/ACM SIGMM MMCN*, Jan. 2000.

Longzhe Han is a Ph.D. candidate in the Dept. of Computer Science at Korea University, Seoul, South Korea. He received the M.S. degree in computer software from Myongji University, in 2006. His research interests are multimedia communications, wireless networks and embedded software engineering.

Sungjun Park received his B.S in material science and engineering from Korea University in 2002, and M.S. in Computer Science from Korea University in 2009. Currently, he is a senior engineer at Innoace, Co.,Ltd. His research interests include mobile live p2p streaming, scalable video coding, and mobile platform.

Seung-Seok Kang received his B.Sc. degree in Computer Science from the Korea University in 1992, and his M.S. degree and Ph.D. in Computer Science and Engineering from Michigan State University, East Lansing, Michigan, in 1998 and 2004, respectively where he worked as s research and teaching assistant. Currently, he is with the Seoul Women's University in Seoul, Korea (ROK) as an assistant professor. His research interests include the area of ad hoc networks, anonymous communications, content sharing, multimedia communications, QoS, P2P networks, and mobile computing.

Hoh Peter In is an Associate Professor in the Dept. of Computer Science at Korea University at Seoul, South Korea. His primary research interests are embedded software engineering, social media platform and service, and software security management. He earned the most influential paper award for 10 years in ICRE 2006. He published over 100 research papers. He was an Assistant Professor at Texas A&M University. He received his Ph.D. in Computer Science from University of Southern California (USC).