Australian Journal of Basic and Applied Sciences, 8(2) February 2014, Pages: 371-380

AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Dynamic Redundancy FEC for Supporting Video Delivery over IP Networks

[1]Radwan Abujassar, [2]Omar Almomani, [3]Albara Awajan, [3]Jafar A. Al-Zubi

[1]*Middle East University, 1Dept. of Computer Information System, IT fauclty,Amman. Jordan.*
[2]*Department of Network and Computer Information System, The World Islamic Sciences & Education University, Jordan.*
[3]*Dept. of Computer System Engineering,Faculty of Engineering, Al-Balqa Applied University. Al-Salt. Jordan.*

ARTICLE INFO

Article history:
Received 1November 2013
Accepted X January 2014
Available online February 2014

Keywords:
FEC,DRFEC,Video Quality,Network performance,NS2

ABSTRACT

Background: *This paper proposes a Dynamic Redundancy Forward Error Correction (DRFEC) mechanism, implemented in the end devices (Receivers). Unlike traditional static FEC mechanisms, which add redundant data to transmission data in a fixed number, the number of redundant FEC data for the DRFEC mechanism is determined by the receiver, which is* based on packet gap sequence number and time-out, *The design goals of the mechanism are to enhance the video streaming quality over existing IP network by reconstructing loss packets and to enhance network performance by minimizing delay and consumed bandwidth. The proposed mechanism is implemented in simulation environment using the NS2 network simulation. The performance analysis and simulation experiments showed that our proposed mechanism performs better in comparison with the traditional static FEC mechanisms. The results showed that, using the DRFEC mechanism can decrease the consumed bandwidth and can also decrease the delay when compared with the traditional static FEC mechanism. Therefore, based on the findings of this study, using DRFEC is a potentially viable mechanism of improving the network performance and video quality.*

To Cite This Article: Radwan Abujassar, Omar Almomani, Albara Awajan, Jafar A. Al-Zubi., Dynamic Redundancy FEC for Supporting Video Delivery over IP Networks. ***Aust. J. Basic & Appl. Sci.,*** *8(2): 371-380, 2014*

INTRODUCTION

Video streaming applications over the Internet is suffering many challenges, packet loss is one of the main challenges Apostolopoulos (2002), Feamster and H. Balakrishnan (2002). This is a result of best-effort services provided by existing IP networks, which does not guarantee packet delivery. Therefore, Forward Error Correction (FEC) is a mechanism used to alleviate the effect of packet losses in the Internet by adding fixed extra packets known as parity packets or redundant packets, which are used to reconstruct the original packets in the event of losses A. Inoie, Xing Zheng, A. Bouabdallah (2013,2012,2002). The use of redundant packet resulted in more consumed bandwidth and increased end-to-end delay.

FEC requires redundant packets to be added to original video packets to repair the lost packets. Currently, the widely used mechanism in the Internet is static FEC J. Lacan, J. Lacan, L. Vicisano (2009,2007,1999) where redundant packets are added to original packets as a fixed number. The static FEC has some major drawbacks because it reduces network performance by increasing consumed bandwidth and delay due to the fixed number of redundant packets, which is not flexible enough to adapt to network condition variations. This paper aim to propose and tested a mechanism to solve the problem of fixed redundant packets by using a dynamic approach that can adjust to the number of redundant packets dynamically.

The remainder of this paper is organized as follows. The related work is discussed in section 2. The proposed DRFEC concept and algorithm is introduced in section 3. Section 4 discusses simulation experimental settings and results. Finally, concluded the paper in section 5.

Related Works:

The main problem of static FEC is the use of a fixed number of redundant packets. Therefore, several mechanisms have been proposed to solve this problem by adapting the amount of redundant packets based on several techniques, such as was explored in previous studies J. Bolot, C. Lin, Ming-Fong, K. Park, C. Lin, K. French (1999,2008,2006,1998,2001). Here, we will explain each mechanism and find out the deficiencies that need to be improved.

Corresponding Author: Radwan Abujassar, Middle East University, Dept of Computer Information System, IT fauclty, Amman. Jordan.

Park and Wang (2006) proposed the adaptive FEC (AFEC) mechanism for improving end-to-end transport of real-time traffic by adjusting the FEC rate in accordance with feedback information relating to the current network delay.

Meanwhile, Bolot *et al*. (1999) proposed the Adaptive FEC mechanism for adjusting the redundancy in packets based on certain constraints on the total sending rate. Their proposed mechanism achieves close to audio-specific subjective quality, and it provides good performance with single processing.

Another research by French and Claypool (1998) proposed an adaptive FEC mechanism that achieves the minimal end-to-end delay and low loss rates after repair. The proposed mechanism uses media specific FEC that adjusts to the current network burst loss rate.

Further work was performed by Lin *et al*. (2006) who proposed the Enhanced Adaptive FEC (EAFEC) mechanism for video delivery over a wireless network. The number of redundant data is determined by Access Point (AP) based on the network traffic load and wireless channel status. They used queue length to indicate the network traffic load and packet transmission time to indicate wireless channel status. They fixed four thresholds: one for low queue length, one for high queue length, one low retransmission time, and one for high retransmission time.

In another vein, Tsai *et al*.(2008) proposed Burst-aware Adaptive FEC (BAFEC) for improving video streaming over a wireless network. BAFEC takes into account burst packet loss length, and gives feedback to the sender regarding the average burst packet loss length, so that the sender can determine the amount of FEC redundancy.

Meanwhile, Lin *et al*. (2006) proposed a novel solution to improve the quality of video delivered over WLANs. Unlike previous AFEC schemes in which the FEC rate is determined based on feedback information supplied by the receiver side, an appropriate FEC rate is determined by the wireless access point based upon an assessment of the current network traffic load. In comparison with previous solutions, the proposed scheme brings significant performance improvements without injecting too many redundant packets into the network.

From the literature, we can see the mechanisms in previous work J. Bolot, C. Lin, Ming-Fong, K. Park, C. Lin, K. French (1999, 2008, 2006, 1998, 2001) aimed to improve video streaming over wireless network environments. The work reported in this paper is designed to implement the dynamic redundancy FEC mechanism in order to improve video streaming over the Internet. The mechanism dynamically determines the amount of redundant FEC packets that should be generated, based on the information feedback from the receiver.

The comparison of different FEC mechanisms proposed to adjust the redundancy packet amount have been summarized in Table 1. We shall propose an intelligent FEC redundancy mechanism adjustment for improving video streaming quality and network performance over the Internet, that operates based on best effort services, which have no quality of service guarantee. The proposed mechanism differ from other previous mechanisms J. Bolot, C. Lin, Ming-Fong, K. Park, (1999,2008,2006), which are used to improve video streaming quality and network performance for a wireless environment, meanwhile. It is also differ from the mechanism as J. Bolot (1999) indicated that is used for improving audio quality, where the proposed mechanism is used to improve the video quality for wire networks and network performance. The proposed mechanism determines the number of redundant data based on information feedback transmitted from the receiver to sender, which is not like the mechanism J. Bolot(1999) that determines the number of redundant data by access points. Then we do not use queue length and packet transmission time as in the other work, and average burst packet loss length previously as indicators to adjust the redundancy packets. In this paper, we used the number of lost packets of the block to adjust the number of redundancy packets. The mechanism calculates the amount of lost packets per block when the block is totally received by the receiver, then the receiver feedback the sender the number of lost packets in the block.

Table 1: Comparison of Different FEC Mechanisms

FEC Mechanism	Techniques	Disadvantage
AFEC	Adjust redundancy packets based on the network state indicated by FEC rate in accordance with feedback information relating to the current network delay	This mechanism measures delay, and then calculates to return an appropriate FEC rate, which inevitably has a finite duration, and the FEC rate implemented at the sender may not accurately reflect the current network load
Adaptive FEC	Adjust redundancy packets based on certain constraints on total sending rate	This mechanism achieves close to audio-specific subjective quality only.
Adaptive FEC	Adjust redundancy packets based on the burst loss rate condition	This mechanism added delay, this study does not consider delay as tolerable by the application
EAFEC	Adjust redundancy based on network condition indicated by traffic load (queue length) and wireless channel state (retransmission time)	This mechanism does not consider the effect of packet loss from wired network on video recovery, and there were not any optimum threshold values
BAFEC	Adjust redundancy packets based on packet loss rate and burst packet loss length	This mechanism does not consider the effect of packet loss from wired network on video recovery and although the mechanism overcomes the burst packet loss over wireless

		network, the packet loss rate cannot give any indication of channel burst packet loss
RED-FEC	Adjust redundancy packets based on network condition indicated by queue length at the access point	This mechanism does not account for the effects of packet loss from wired network on video recovery, but this research has fixed the lower and upper threshold values that were not set to optimal.

Dynamic Redundancy FEC (DRFEC) mechanism:
2.1 Theoretical Model of DRFEC:

The theoretical modeling technique is a set of equations describing the performance of a computer network system, which is expressed using mathematical symbolism to represent an actual computer network system; this might lead to a better decision of a system before the implementation process; and the steps include building, solving, and validating the analytical model to explore and solve the problem. This technique is used to study simple systems; if this technique is used to study complex systems, it would require simplification and assumptions and this is not easy A. Law (1991). The loss probability of video packet calculated is given by J. Roberts (2001):

$$\pi(\rho) = \frac{1-\rho}{1-\rho^{l+1}}\,\rho^{l} \tag{1.1}$$

and for $\rho = 1$,

$$\pi(\rho) = \frac{1}{l+1} \tag{1.2}$$

We added redundancy to each packet in a way that if a packet is lost. So the loss probability of a video packet becomes.

$$\pi(\alpha) = \frac{1-N\rho/K}{1-(N\rho/K)^{l+1}}\left(\rho\frac{N}{K}\right)^{l} \tag{1.3}$$

The receiver can recover all lost packets of the R-S codeword set of N packets as long as the number of lost packets does not exceed (N-K), which are the redundant packets. So if the redundant packets are less than the ones lost, this will produce bad quality, and if redundant packets are more than the lost ones, this will worsen network performance by sending unnecessary packets. Therefore, adapting the amount of redundant packets based on receiver feedback control can improve quality and network performance. Since the main goal of feedback control [17] is to make output equal to input, the common setup for feedback control is illustrated in the following Figure 5.1:

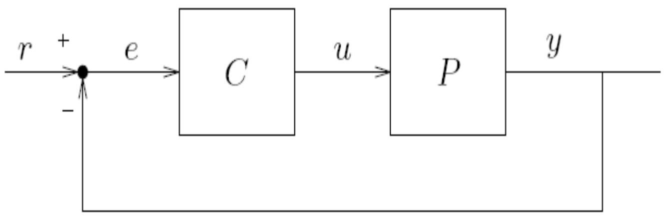

Fig. 1: Feedback Control

Where:
P is the plant,
C is the controller,
u is the plant input,
y is the plant output,
r is the reference or command input, and
e is the error (drop packet).

$$e= r-y \tag{1.4}$$

Where:

$$R= N-K \tag{1.5}$$

R is the redundant packet,
N is the length of the block, and
K is the original volume of packets.

The redundant packets equal the error (drop packet), and therefore:

$$R=e \tag{1.6}$$

So the block length is given by:

$$N= K + r- y \tag{1.7}$$

By substituting 1.7 into 1.3, we can calculate the loss probability of video packets for FEC (N,K) with feedback control. Therefore, the receiver can recover all lost packets of the R-S codeword set of N packets regardless of the number of lost packets.

Packet Format of DRFEC:

DRFEC is run in the top of UDP, UDP data units are called datagrams. Each UDP datagram is composed of a header and a payload (user data). In the payload the data coming from the layer above is encapsulated. Figure 2 shows the header structure of UDP.

0 16 31

Source Port	***Destination Port***
Length	***Checksum***

Fig. 2: UDP header

The fields source port and destination port have 16 bits each. They identify the source computer application and destination computer application respectively. The length field has 16 bits. It indicates the UDP datagram length including the header. The checksum field is an optional field, it protects the header and the data. Figure 3 shows how the data from the application is encapsulated as it is lowering in the UDP/IP protocol stack.

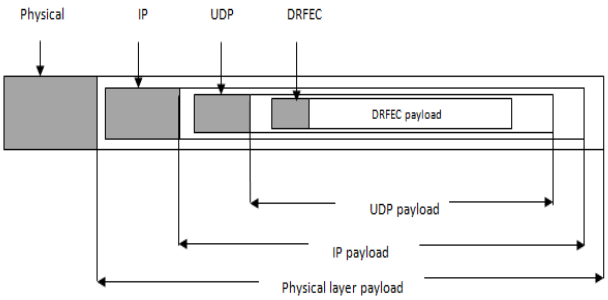

Fig. 3: Protocol Encapsulation

FEC is independent of the nature of the application data. That means that a FEC packet is obtained by placing the FEC header and the FEC payload in the UDP payload, as it is shown in figure. 4 The FEC payload is composed by the application data and the FEC header is constructed by placing on it the redundant packets and block length.

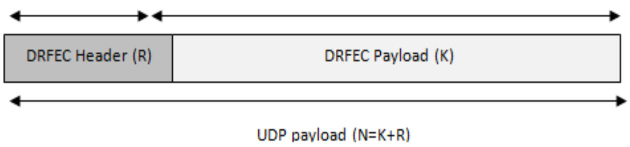

Fig. 4: DRFEC Packet Format and its Encapsulation on UDP Protocol

Australian Journal of Basic and Applied Sciences, 8(2) February 2014, Pages: 371-380

DRFEC mechanism algorithm:

Dynamic redundancy FEC requires a distinct sender and receiver agent. The sender is the one that updates the number of redundant packets based on receiver feedback, while the receiver will determine the amount of lost packets in the block based on the gap sequence number and time out, which are the most common loss detection techniques for streaming applications A. Argyriou, I. Kofler (2008). Then, the receiver sends the feedback message to the sender, and this message contains the number of lost packets and block number, i.e. (Notification Blk:2,Loss:5). This means in block number two, five packets had been lost. Figure 5 shows the DRFEC algorithm pseudo code.

```
Begin
Input N, K, R, Loss
IF Loss > 0 then
R=Loss
N=K+R
set FEC(N,K)
ELSE
R=0
N=K
set FEC(N,K)
END
```

Fig. 5: DRFEC pseudocod

1DRFEC Sender Agent Operation:

DRFEC sender waits for the request from DRFEC receiver for the packets. After the request reception, the sender has the responsibility to encode original packets plus redundant packets into one block. The total number of packets sent per block is N=K+R, where N is block length and its value determines by administrator, K is the original packet and R redundant packet and its value determines based on the loss packets. When the blocks of packets are received on the receiver side, the receiver detects and calculates the lost packets based on gap sequence number and time-out, and then sends the feedback information message of lost packets per block to the sender. Figure 6 shows the basic DRFEC sender and receiver agent operation.

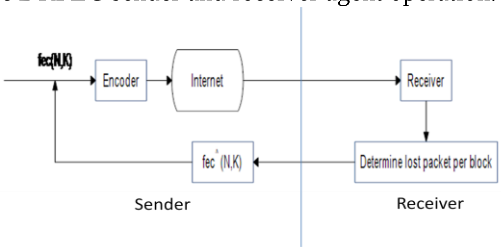

Fig. 6: DRFEC Sender and Receiver Agent Operation

After the sender received a feedback message, it updates the number of redundancy packets. The DRFEC sender agent algorithm can be summarized as follows:

i. Sender waits for request from receiver.
ii. Sender encodes n,k in the block following n=k+r, where is r= 0.
iii. Sender sends first block and once completed, sender does not wait until received feedback message before the next block starts to be sent.
iv. Sender receives feedback information about completed block, after that adjusting amount of redundant packets where r= loss packet, the sender resends lost packets with number of block.
v. Finally, sender completes resending of lost packets.

2.3.2 DRFEC Receiver Agent Operation:

The DRFEC receiver sends a request to the DRFEC sender. The receiver then makes tracks to ensure that the packets in the block are received. After the block is totally received, the receiver checks the number of lost packets. Lost packets are determined based on packet gap sequence number and time-out, which is the most common loss detection technique used for streaming applications H. Sze and N. Feamster (2001&2002). After that, the receiver updates the information and sends the feedback information to the sender. This feedback information is the number of lost packets. Finally the sender generates a redundant packet that is equal to the lost packets. DRFEC receiver agent algorithm can be summarized as follows:

i. The DRFEC receiver sends a request to the DRFEC sender.
ii. After DRFEC receiver receives the first block, it requests for the next block.
iii. DRFEC receiver starts to make tracks for received block to calculate the packet lost; first check the gap sequence and if found the gap sequence to be large, then check time-out.
iv. After DRFEC receiver completed the checking and determined the amount of lost packets, it sends notification message to sender using the following format, (Notification Blk: 2, Loss: 5).

Experimental Design and Results:

This experiment is to evaluate the DRFEC with Drop Tail queue policy for varying queue sizes to reconstruct packets loss, and compare DRFEC with static FEC. Selection of appropriate network topologies in simulating communication network systems is very important. The right network topology ensures that it is representing the problems under investigation, and the simulation results are as general as possible. In this paper, the experiment for analyzing FEC and DRFEC were conducted by mean of extensive simulation using single-bottleneck topology. Figure 7 shows the general view of single-bottleneck simulation topology used in this paper. This topology was also used by other researchers in their study J. Bolot, C. Lin, Ming-Fong, K. Park, C. Lin, K. French (1999,2008,2006,1998,2001) to analyze, study and evaluate FEC over the wired and wireless network.

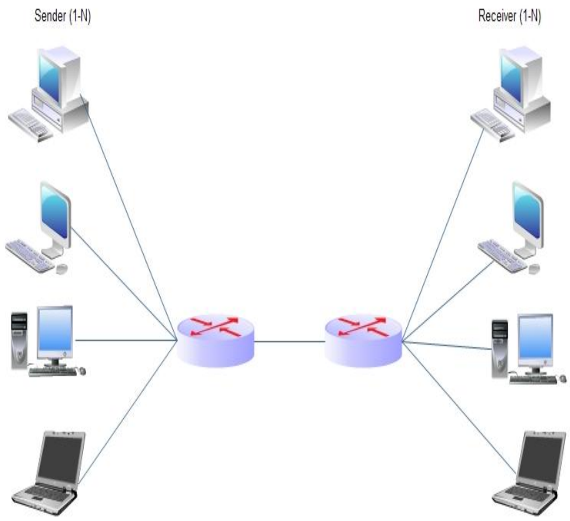

Fig. 7: Single-Bottleneck Simulation Topology

i. The bottleneck bandwidth is shared by a DRFEC, FEC, and TCP. We used the duplex link to link up all the connections. The duplex link enables packets to flow in both directions from sender to receiver and vice versa. We used the CBR for video traffic because it closely represents the behavior of real video data, and attached it to FEC, DRFEC, and we used competing TCP traffic (FTP) flow to increase the packet losses. The basic setting and parameters used in this experiment are shown in Table 2.

Table 2: Simulation Parameters

Simulation Parameter	Scenario
Queue Policy	Drop Tail
Queue Size	20,40,60,80, and 100
FEC Block number	30
FEC Block size	255
Simulation time	400s

Table 3 shows the number of loss packets, redundant packets, received packets, bandwidth and delay of the DRFEC with DT queue policy implementation in the environment with varying queue sizes (20, 40, 60, 80, and 100).

Table 3: DRFEC

Queue Size	Loss	Redundat	Received Packets	Bandwidth (kbps)	Delay (ms)
20	230	230	6690	1916	356
40	174	174	6690	1916	436
60	142	142	6690	1916	452
80	110	110	6690	1916	542
100	88	88	6690	1916	555

While each row represents the results from distinct simulations, i.e. queue size = 20, the number of loss packets amounted at 230 and it is equal to the number of redundant packets. Meanwhile the received packets totally 6690 and they are equal to the original data packets (30*255=6690), while the bandwidth is 1916kbps and delay is 356ms.

By increasing the queue size, i.e. 40, the number of loss has changed to 174 and it is equal to number of redundant packets, while the received packets have become 6690 and it is equal to original data packet (30*255=6690). Finally the bandwidth has become 1916kbps and the delay is 436ms.

Finally, we can generalized that, by increasing the queue size, the number of loss decreased and it is equal to number of redundant packets, while the received packets stayed at 6690 and it is equal to original data packet (30*255=6690); the bandwidth is stayed fixed for different queue size.

Table 4 shows the number of loss packets, redundant packets, received packets, bandwidth, delay, and inept packets of the FEC with DT queue policy implementation in the environment with varying queue sizes (20, 40, 60, 80, and 100). Meanwhile each row represents the results from distinct simulations i.e. queue size = 20, the number of loss was 237, the number of redundant packets was 450 (15*30), and the received packet was 6903, whereas the original data packets was 9960 (30*255=6690), the bandwidth was 1977kbps, the delay was 518ms, and the extra packet sent through network was 213.

Table 4: FEC

Queue Size	Loss	Redundant	Received Packets	Bandwidth (kbps)	Delay (ms)	Extra Packet
20	237	450	6903	1977	518	213
40	187	450	6953	1992	534	263
60	156	450	6984	2001	551	294
80	117	450	7023	2012	609	333
100	96	450	7044	2018	647	353

By increasing the queue size to 40, the number of loss reduced to 187, number of redundant packets was still 450 (15*30), the received packets increased to 6953; whereas the original data packets were still 6690 (30*255=6690), the bandwidth increased to 1992kbps, the delay increased to 534ms, and the extra packets sent through network increased to 263.

By increasing the queue size, i.e. 60, the number of loss reduced further to 156, the number of redundant packets remained at 450 (15*30), the received packets increased further to 6984; whereas the original data packets remained as 6690 (30*255=6690), the bandwidth increased further to 2001kbps, the delay increased further to 551ms, and the extra packet sent through network increased further to 294.

By increasing the queue size again i.e. 80, the number of loss dwindled to 117, the number of redundant packets stayed at 450 (15*30), the received packets 7023l, whereas the original data packets stayed at 6690 (30*255=6690), the bandwidth jumped to 2012 kbps, the delay rised to 609 ms, and the extra packet sent through network increased to 333.

Finally, by increasing the queue size again to 100, the number of fell to 96, the number of redundant packets was still 450 (15*30), the received packets stopped at 7044; whereas the original data packets were still 6690 (30*255=6690), the bandwidth rised to 2018kbps, the delay increased to 547ms, and the extra packets sent through network increased further to 353.

Figure 8 shows the trends of DRFEC and FEC performance with varying DT queue sizes (20, 40, 60, 80, and 100). From the figure, one can observe that the amount of packet loss of DRFEC is less than the FEC, the reason is that with DRFEC, the sending packet is less than the FEC, i.e. when queue size is 20, the number of sending packet with DRFEC is 6920, while the number of sending packets with FEC is 7140. Here, sending packets is the sum of the original packets and redundant packets.

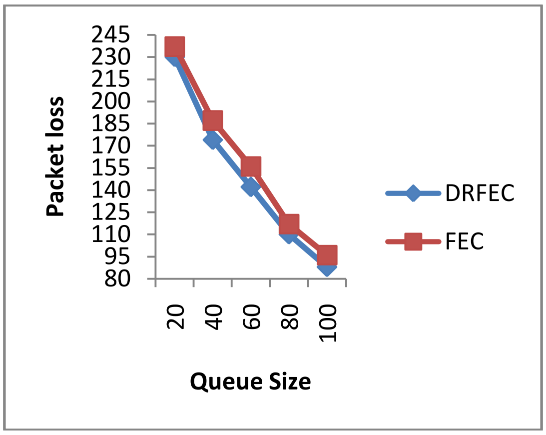

Fig. 8: Packet loss versus Queue Size

It can also be observed that when the queue size is increased the loss packets would decrease for DRFEC and FEC. That is due to the large queue size that can absorb more packets in the queue buffer, which leads to fewer packets loss.

Figure 9 shows the trends of DRFEC and FEC performance with varying DT queue sizes (20, 40, 60, 80, and 100). From the figure, it can be observed that the amount of packet redundancy of DRFEC is less than the FEC. The reason is due to the fact that in FEC a fixed number of redundant packets is added; whereas in DRFEC the redundant packets are added based on feedback from the receiver with the number of lost packets. Therefore, when queue size is increased, the number of packets lost is decreased, so the number of redundant packets are decreased as well.

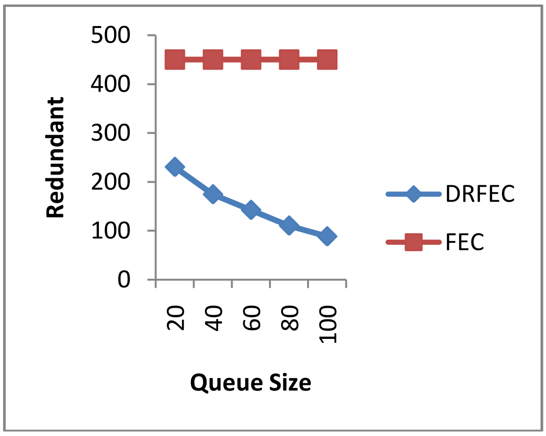

Fig. 9: Redundant Packet versus Queue Size

Figure 10 shows the trends of DRFEC and FEC performance with varying DT queue sizes (20, 40, 60, 80, and 100). From the Figure 10, it can be observed that the required bandwidth of DRFEC is less than the FEC. The reason is because the FEC sends more redundant packets than DRFEC. From the figure also, it can be observed that the bandwidth was not affect by varying the queue size for DRFEC. The reason for that is the DRFEC redundant packets are added based on the receiver feedback. When the queue size was 20, the lost packets were 230, and redundant packets were 230. Meanwhile, when the queue size was 100, the lost packets were 88, and redundant packets were 88. On the other hand, the FEC sends a fixed number of redundant packets (450), thus requiring more bandwidth. For example, when the queue size was 20, lost packets were 237, and redundant packets were 450, which meant that 213 extra packets were sent through network and these packets consumed more bandwidth as compared to DRFEC. Therefore, DRFEC is more friendly to network performance than FEC.

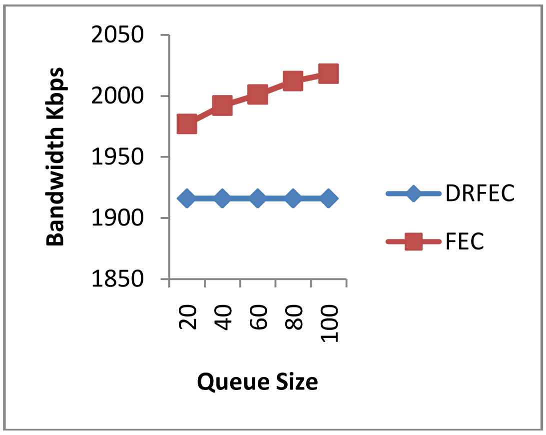

Fig. 10: Bandwidth versus Queue Size

Figure 11 shows the trends of DRFEC and FEC performance with varying DT queue sizes (20, 40, 60, 80, and 100). From the Figure 6.8, it can be observed that using DRFEC produced less delay than FEC. The reason is that the FEC sends more packets than the DRFEC.

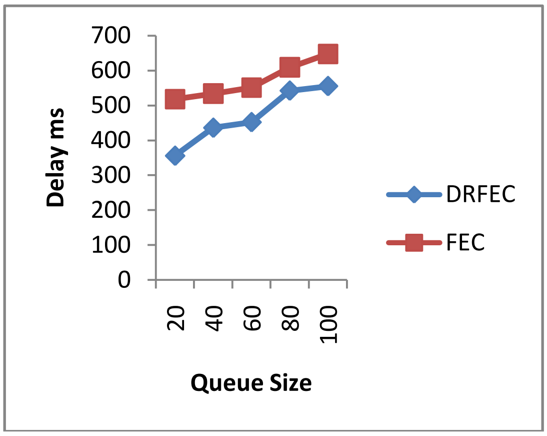

Fig. 11: Delay versus Queue Size

Conclusions:

DRFEC mechanism has been proposed, designed, and implemented. This mechanism can be adjusted by the appropriate number of redundant packets depending on feedback from the receiver; the redundant packets are determined by the gap sequence and time out for each block. Moreover, this mechanism can be easily implemented in the current Internet infrastructure. This mechanism has been evaluated using several scenarios. The results from simulation experiments showed that the DRFEC mechanism performs better in comparison to the static FEC.

REFERENCES

Apostolopoulos, J., W. Tan and S. Wee, 2002. "Video Streaming: Concepts, Algorithms, and Systems," Mobile and Media Systems Laboratory,HP Laboratories, Palo Alto, CA USA report HPL-2002-260.

Feamster, N. and H. Balakrishnan, 2002. "Packet Loss Recovery for Streaming Video," presented at 12th International Packet Video Workshop, Pittsburgh,USA.

Inoie, A., 2012. "Audio Quality in Lossy Networks for Media-Specific Forward Error Correction Schemes," International Journal of Communication Systems.

Chan, S.-H.G., Xing Zheng, Qian Zhang, Wen-Wu Zhu and Ya-Qin Zhang, 2006. "Video Loss Recovery with FEC and Stream Replication," IEEE Transactions on Multimedia, 8: 370-381.

Bouabdallah, A., M. Cunche, V. Roca, K. Matsuzono and J. Lacan, 2013. "Simple Reed-Solomon Forward Error Correction (FEC) Scheme for FECFRAME," *Internet Engineering Task Force (IETF) Request for Comments 6865*.

Vicisano, L., M. Watson and M. Luby, 2007. "Forward Error Correction (FEC) Building Block," *RFC 5052, Internet Engineering Task Force*, http://www.ietf.org/rfc/rfc5052.txt.

Lacan, J., V. Roca, J. Peltotalo and S. Peltotalo, 2009. "Reed-Solomon Forward Error Correction (FEC) Schemes," *RFC 5510, Internet Engineering Task Force*, , http://www.ietf.org/rfc/rfc5510.txt?number=5510.

Watson, M., 2009. "Basic Forward Error Correction (FEC) Schemes," *RFC 5445, Internet Engineering Task Force*, http://www.ietf.org/rfc/rfc5445.txt?number=5445.

Bolot, J., S. Fosse-Parisis and D. Towsley, 1999. "Adaptive FEC-Based Error Control for Internet Telephony," presented at Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA.

Lin, C., C. Ke, C. Shieh and N. Chilamkurti, 2006. "An Enhanced Adaptive FEC Mechanism for Video Delivery over Wireless Networks," presented at International Conference on Networking and Services, Slicon Valley, CA USA.

Ming-Fong Tsai, Chih-Heng Ke, Tsung-Han Wu, Ce-Kuen Shieh and W.-S. Hwang, 2008. "Burst-aware Adaptive Forward Error Correction in Video Streaming over Wireless Networks," presented at 10th IEEE International Conference on High Performance Computing and Communications, Dalian,China.

Park, K. and W. Wang, 1998. "AFEC: an Adaptive Forward Error Correction Protocol for End-to-End Transport of Real-Time Traffic," presented at 7th International Conference on Computer Communications and Networks, Lafayette, LA,USA.

Lin, C., C. Shieh, N. Chilamkurti, C. Ke and W. Hwang, 2008. "A RED-FEC Mechanism for Video Transmission over WLANs," *IEEE Transactions on Broadcasting*, 54: 517-524.

French, K. and M. Claypool, 2001. "Repair of Streaming Multimedia with Adaptive Forward Error Correction," presented at ITCOM, Denver, CO, USA.

Law, A. and W. Kelton, 1991. *Simulation Modelling and Analysis*, Third ed: McGraw-Hill.

Roberts, J. and F. R&D, 2001. "Traffic Theory and the Internet," *IEEE Communications Magazine*, 39: 94-99.

Argyriou, A., 2008. "Cross-Layer Error Control for Multimedia Streaming in Wireless/Wireline Packet Networks," *IEEE transactions on multimedia*, 10: 1121-1127.

Kofler, I., J. Seidl, C. Timmerer, H. Hellwagner, I. Djama and T. Ahmed, 2008. "Using MPEG-21 for Cross-Layer Multimedia Content Adaptation," *Signal, Image and Video Processing*, 2: 355-370.

Sze, H., S. Liew and Y. Lee, 2001. "A Packet-Loss-Recovery Scheme for Continuous-Media Streaming over the Internet," *IEEE Communications Letters*, 5: 116-118.

Feamster, N. and H. Balakrishnan, 2002. "Packet Loss Recovery for Streaming Video," presented at 12th International Packet Video Workshop, Pittsburgh, PA.

Kang, S.-R. and D. Loguinov, 2007. "Modeling Best-Effort and FEC Streaming of Scalable Video in Lossy Network Channels," IEEE/ACM Transactions on Networking, 15: 187-200, 2007.

Huang, C., S. Sukittanon, J. Ritcey, A. Chindapol and J. Hwang, 2006. "An Embedded Packet Train and Adaptive FEC Scheme for VoIP over Wired/Wireless IP Networks," presented at IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, French.

Alemu, T., Y. Calas and A. Jean-Marie, 2005. "The Interaction of Forward Error Correction and Active Queue Management," presented at IFIP Networking'05, Waterloo, Ontario, Canada.