2009 First International Conference on Networks & Communications

Performance Study of Large Block FEC with Drop Tail for Video Streaming over the Internet

Omar Almomani
Department of Computer Sciences
College of Arts and Sciences
Universiti Utara Malaysia
Sintok, Kedah, Malaysia
almomani81@yahoo.com

Osman Ghazali
Department of Computer Sciences
College of Arts and Sciences
Universiti Utara Malaysia
Sintok, Kedah, Malaysia
osman@uum.edu.my

Suhaidi Hassan
Department of Computer Sciences
College of Arts and Sciences
Universiti Utara Malaysia
Sintok, Kedah, Malaysia
suhaidi@uum.edu.my

***Abstract*- This paper, showed an investigation on performance of the large block forward errors correction (FEC) with Drop Tail (DT) queuing policy. FEC is a technique that uses redundant packet to reconstruct the dropped packet, while Drop Tail is the most popular queue management policies used in network routers. Since the Drop Tail mainly depends on the size of the queue buffer to decide on whether to drop a packet or not, the investigation considered simulation settings with varies size of the queue buffer. Results obtained from the simulation experiments show that FEC and queue size affect the performance the network. Consequently, the qualities of multimedia applications are also affected.**

***Keywords*-Forward Error Correction; Drop Tail; Packet Loss; TCP**

I. INTRODUCTION

The Internet traffics suffer from heavy losses due to network congestion caused by the limited capacity of queues in the routers. A loss refers to a situation where a packet does not arrive at the destination, or arrive at the destination but late that caused it to be unusable. This usually happens when a network is heavily loaded. Congestion in the network is the most common reason for packet loss [1, 2]. This loss decreased the network performance. However, packet loss and large delays in data transmission are often unacceptable.

Error control correction is used to reconstruct the lost data by either retransmission of data from the sender using ARQ (Automatic Repeat request) or by adding redundant data using FEC (Forward Error Correction). FEC is a method of error control correction, used to correct the error in data transmission by adding redundant data at the sender. When a receiver detects an error it will reconstruct the lost data from the redundant data without retransmission of the lost data from the sender. There are several limitations of FEC. That is FEC cannot recover all lost packets. In addition, the transmission of redundant packets increases the overall network load. The effectiveness of FEC is known to depend on the way packet drops are distributed in the data stream. FEC is more efficient when packets losses are independent [3].

Queue policies mechanisms refer to traffic policies techniques at a router that detect and notify traffic sources of imminent network congestion to prevent outbound buffer overflow and control queuing delay [4]. When being notified of network congestion, cooperative traffic sources like TCP reduce their transmission rates to participate in the congestion control. In that case network congestion cannot be managed voluntarily by the traffic sources. Queue policies may use buffer management techniques to suppress traffic to the targeted traffic level and achieve the QoS goal. Traditional Internet routers used Drop Tail queue management [5], which drops the arriving packets if the buffer of the output port overflows.

This paper organized in the following manners. Section 2 provides the description of Drop Tail queue. Section 3 provides the description of FEC mechanism. Section 4 describes simulation experimental set up. Section 5 explains evaluation metric. Section 6 discusses the simulation results. Finally, concludes this paper.

II. DROP TAIL QUEUE POLICY

Drop Tail is a simple queue policy algorithm used in Internet routers, it drops packets base on the queue size. In contrast to the more complex algorithms like RED [6]. Using Drop Tail each packet is treated identically. When the queue is filled to its maximum capacity, the newly arriving packets are dropped until the queue has enough room to accept incoming traffic. The Drop Tail has two disadvantages: namely lock-out and full queue. Lock-out is possible that one or more flows can monopolize the queue in the router. Full queue is dropped packet only when the queue currently full. The solution to the full-queues problem is for routers to drop packets before a queue becomes full. By dropping packets before buffers overflow, active queue management allows routers to control when and how many packets to drop.

Figure 1 illustrate Drop Tail's infrastructure, drop tail processes the forwarding order of packets depending upon their arrival time.

978-0-7695-3924-9/09 $26.00 © 2009 IEEE
DOI 10.1109/NetCoM.2009.12

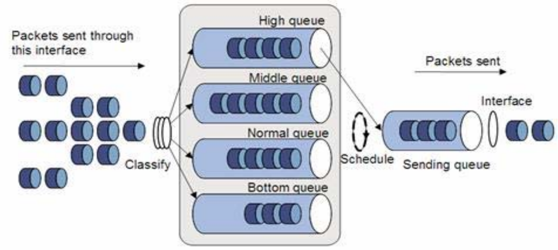

Figure 1. Drop Tail's Infrastructure[1]

III. FEC MECHANISM

Forward error correction (FEC) has been proposed to recover packets losses in real time applications, i.e. audio and video applications, based on the redundant information. A number of forward error correction techniques have been developed to repair losses of data during transmission [7-9]. FEC enables the receiver to correct losses without dealing with the sender.

Forward Error correction sends original and redundant data as a block of FEC (n, k), where k is the number of data packets in a FEC block and n is the number of all the packets in the FEC block.

$$R=\frac{k}{n} \tag{1}$$

R is the encoding rate of block. Codes that introduce less redundancy have higher code rates, and transmit more information per code bit.

There are two approaches of FEC design to recover data packets from losses - media dependent or media-specific and media independent. Media dependent FEC works against packet loss by transmitting each packet more than one time. When a packet is lost, one of its extra packets is able to restore it. The first packet transmitted of audio or video is the main encoding because it has the best quality. Duplicates of this packet is the minor encoding because the sender is able to decide if the quality or bandwidth of this packet should be the same or lower than the main encoding packet. Media- independent FEC does not need to know what is inside the contents of the stream. Block or algebraic codes are transmitted to help repair what was lost. There are k data packets in a codeword and n-k extra check packets are transmitted for n packets that need to be sent over the Internet. Figure 2 illustrate media-independent FEC.

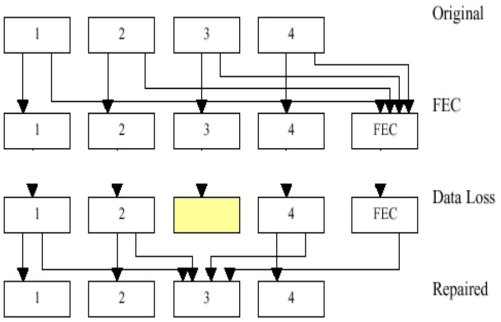

Figure 2. Media-independent FEC

FEC Codes are commonly divided into two classes, block codes and convolutional codes. Block codes are based on finite field arithmetic and abstract algebra. They can be used to either detect or correct errors. Block codes consist of a block of K information bits producing a block of n coded bits. By encoded rules, n-k redundant bits are added to the k information bits to form the n coded bits. Generally, these codes are referred to as (N, K) block codes. Some of the regularly used block codes are Hamming codes, BCH codes, and Reed Solomon codes [10].

IV. EXPERIMENTAL SETUP

To study the effects of different buffer size of the TD queue policy with FEC on network performance, we conducted a simulation for single bottleneck topology (dumbbell) as illustrate in Figure 3.

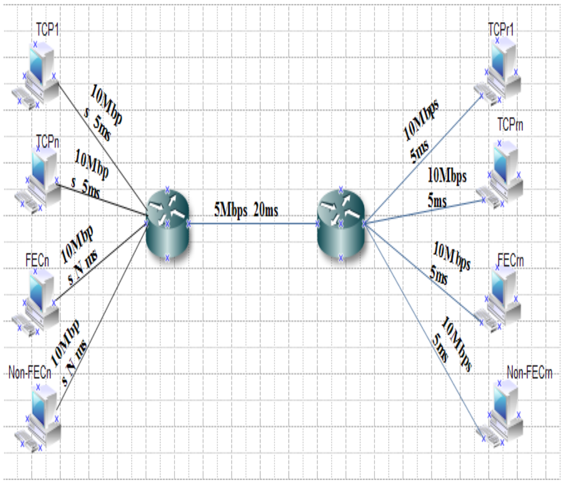

Figure 3 Simulation Topology

FTP traffic attached to TCP sources, CBR traffic attached to FEC and Non-FEC sources. The FEC and Non-FEC sources connected to R1 with a bandwidth of 10Mbps and delay was generated randomly using uniform distribution to achieve the heterogeneous environment of Internet, We used the CBR model in ns-2 for video traffic because it closely represents the behavior of real video data. These configurations cause congestion and loss at R1 so we can identify the efficiency of the FEC to recover packet losses based on

---

[1] http://www.h3c.com/portal/Products___Solutions/Technology/QoS/Technology_Introduction

received packets at the receiver. Table 1 shows our simulation parameters.

Table 2. Simulation Parameters

Parameter	**Value**
Simulation technique	Ns2
Competing traffic	TCP & CBR
Bottleneck bandwidth	5 Mbps
Bottleneck delay	20 ms
Sidelink bandwidth	10 Mbps
Queue Size	20,40,60,80 and 100
FEC block size	255
FEC redundant packets	15
Simulation time	100 seconds

V. PERFORMANCE EVALUATION METRICS

Selecting a performance evaluation metric is a key step and an important part in all performance evaluations. In this paper we used the following metrics:

A. End-to-end delay

$$D = Td - Ts \tag{2}$$

Where Td is the packet receives time at the destination and Ts is the packet sends time at the source node.

B. Bandwidth

$$\text{B.W=} \frac{Rp * pS * 8}{OT} \tag{3}$$

Where Rp is received packet, pS is packet size and OT is observed time.

C. Packet loss

$$Pd = Ps - Pa \tag{4}$$

Where Ps is the amount of packets sent and Pa the amount of packets received.

D. Throughput

$$Tp = \frac{Pa}{Pf} \tag{5}$$

Where Pa is the packets received and Pf is the amount of forwarded packets over a certain time interval.

VI. SIMULATION RESULTS

Simulations were run for 30 times. A random number generator is used to randomly generate the starting time a traffic flow. The experiment was run for 100 seconds. The first 20 seconds was ignored due to instability of the simulation in initial start up. The results were presented with a 95% confidence interval.

Tables 2 and 3 present the results of simulation experiment. Table 2 shows the results of using FEC

Table 1. Using FEC

Queue Size	Total Loss	Total Received	Bandwidth Kbps	Delay Ms
20	237±11	6903	1977±3.17	518±116
40	187±18	6953	1992±5.21	534±123
60	156±9	6984	2001±2.64	551±113
80	117±2	7023	2012±0.82	609±121
100	96±9	7044	2018±2.72	647±108

Table 3 Without Using FEC

Queue Size	Total Loss	Total Received	Bandwidth Kbps	Delay ms
20	223±9	6467	1852±2	339±101
40	158±10	6532	1871±3	378±121
60	134±8	6556	1887±2	408±95
80	104±0.5	6586	1895±0.15	453±114
100	80±5	6610	1913±1	509±117

Figure 4 showed the throughput at the receiver with various queue sizes with and without using FEC. We can observe that using FEC higher throughput can be achieved compare to without using FEC. This is can be associated to the added redundant data with the FEC block. As been mentioned before, the FEC used redundant packets to reconstruct lost packets at the receiver. Figure 4 also shows that as the queue size is increased the received packets increase. Large queue sizes results in higher throughput and good quality

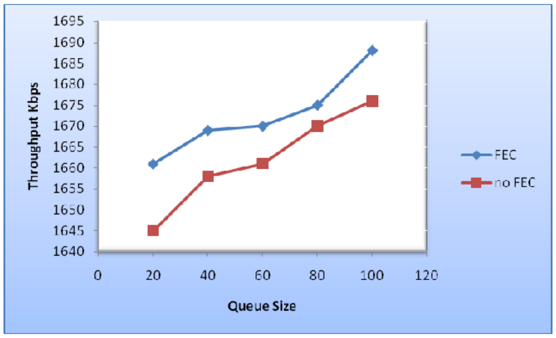

Figure 4 Throughput Vs Queue size

Figure 5 showed the amount of packet loss for various queue sizes with and without using FEC. Through the figure, we can observe that the amount of packet loss with FEC is more than without FEC. The reason is that with FEC we have sent redundant packets, these redundant packets increased the sending packet that caused congestion at the router, so the amount of packet loss increases. But still we can reconstruct the lost packets if the lost packets are less than the redundant packets. For example, with the queue size equal to 60 the amount of packet loss with FEC is 156 and the number of redundant packets equals 450, so we can still reconstruct all the lost packets. But without FEC the amount of packet loss is 134. These lost packets cannot be reconstructed and this caused bad quality. Also, we can observe that when the queue size increased the packet loss decreased. This is because when the queue size is large it allows for more packets to wait in the queue before the router starts dropping any packets.

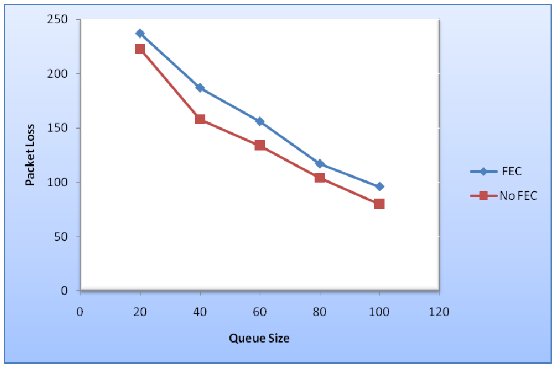

Figure 5 Packet loss Vs Queue size

Figure 6 showed the utilization of bandwidth with various queue sizes with and without using FEC. We observed that by using FEC more bandwidth is required. It is because extra bandwidth is used for the redundant packets. It is justify if there are many packet loss in the network, but waste of bandwidth when the network does not lose packets, i.e. when queue size is 60 the lost packets is 156 and redundant packets is 450, so 156 lost packets will reconstruct from 450 redundant packets. That means 294 useless packets were sent through the network and these packets required bandwidth to transfer. When the queue size is increased the bandwidth is increased. Because when the queue size increased the lost packets decreased, so redundant packets increased.

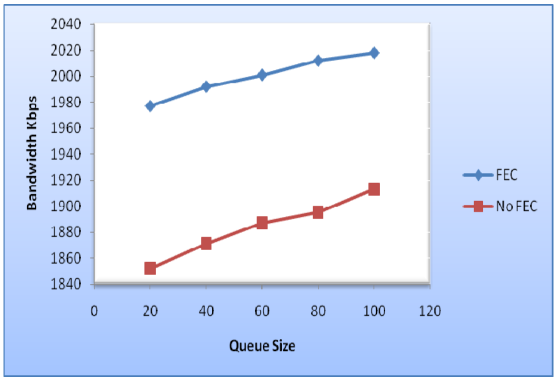

Figure 6 Bandwidth Vs Queue size

Figure 7 showed the end-to-end delay with various queue sizes with and without using FEC. We can observe that using FEC increased the end-to-end delay. This is because of the redundant packets. We can see that when queue size increased the delay increased. This is because the packets stay a long time waiting in the queue before they were forwarded from the router to the other part of network.

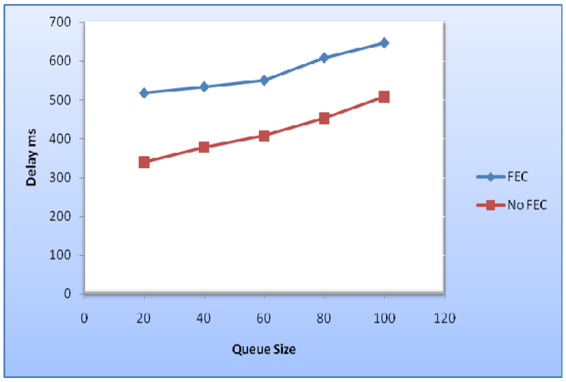

Figure 7 Delay Vs Queue size

VII. CONCLUSION

This paper has been investigated the performance of large block FEC with Drop Tail for different queue size. The investigation was carefully conducted using simulation. From the experiments, the followings are concluded: FEC decreases the network performance because FEC required more bandwidth and increased end- to-end delay as a result of redundant packets. However from user applications perspective, FEC is useful because it can reconstruct lost packet and provided high throughput. Therefore, carefully choosing queue buffer size and redundant packets can improve the performance of video streaming.

REFERENCES

[1] I. Cidon, A. Khamisy, and M. Sidi, "Analysis of Packet Loss Processes in High-Speed Networks," *Information Theory, IEEE Transactions on*, vol. 39, pp. 98-108, 1993.

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

[3] T. Alemu, Y. Calas, and A. Jean-Marie, "The Interaction of Forward Error Correction and Active Queue Management," *Lecture Notes in Computer Science,Springer*, vol. 3462, pp. 1146-1155, 2005.

[4] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, and C. Partridge, "RFC2309: Recommendations on Queue Management and Congestion Avoidance in the Internet," *Internet RFCs*, 1998.

[5] C. Brandauer, G. Iannaccone, C. Diot, T. Ziegler, S. Fdida, M. May, and S. Res, "Comparison of Tail Drop and Active Queue Management Performance for Bulk-data and Web-like Internet Traffic," presented at Sixth IEEE Symposium on Computers and Communications, 2001.

[6] S. Floyd and V. Jacobson, "Random Rarly Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on networking*, vol. 1, pp. 397-413, 1993.

[7] M. Podolsky, C. Romer, and S. McCanne, "Simulation of FEC-Based Error Control for Packet Audio on the Internet," *IEEE INFOCOM*, vol. 2, pp. 505-515, 1998.

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

[9] Q. Zhao, C. Li, X. Zhang, and D. A.-L. Liu, Dongtao, "A New FEC Scheme for Real-time Transmission of High Definition Video over IP Network," presented at The International Conference on Intelligent Pervasive Computing, IPC, 2007.

[10] T. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*: Wiley-Interscience, 2005.