

Study on the Effect of Video Traffic on Responsive Flows Using Queue Management Algorithms

¹P. Sankar and ²C. Chellamuthu

¹Faculty of Electronics and Communication, Jerusalem College of Engineering,
Narayanapuram, 600100 Chennai, India

²Faculty of EEE, RMK Engineering College, Chennai, India

Abstract: A simulation environment has been developed to study video/UDP traffic in a network. In this study, the simulation environment is utilized for the performance analysis of responsive flow in the presence of video traffic using queue management algorithms such as Stochastic Fair Queuing (SFQ) and Stochastic Fair Blue (SFB).

Key words: Bins, ns2, SFB, SFQ, simulation environment, India

INTRODUCTION

The non-responsive flows are UDP flows which are aggressive in nature. On the other hand TCP is a responsive flow which has its own back off mechanism. The internet provides a connectionless, best effort, end to end packet service that depends on the congestion avoidance mechanisms implemented at the end-points (like TCP).

However, bursty applications like multimedia traffic can monopolize network bandwidth and can cause responsive flows to suffer. Therefore, it is necessary to protect responsive flows from non-responsive or aggressive flows in order to provide good Quality of Service (QoS) to all flows:

Due to lack of flow control mechanism, non-responsive flows can starve the responsive flows for buffer and bandwidth at the router. Li and Lee (2003) proposed an adaptive queue management scheme to improve RED algorithm for restraining non-responsive flows (Li and Lee, 2003).

A scheme for estimating the proportion of the incoming traffic that is not responsive to congestion at a router is presented by Zhao *et al.* (2004). A novel scheme which effectively realizes queue stability and flow fairness between non-responsive and TCP flows has been described by Kim *et al.* (2010).

A fair scheduling protocol called time shift protocol for high speed networks has been presented by (Cobb *et al.*, 1998). The performance evaluation of Blue queue management algorithms using CBR/UDP traffic has

been carried out by Feng *et al.* (2002). The present traffic flow pattern in the internet indicates that considerable percentage of volume is video/UDP traffic. The survey shows that the performance study of responsive flow in presence of video traffic (non-responsive flow) has not been studied in detail.

The researchers have developed a simulation environment using ns2, ffmpeg codec and its service utilities to study the performance of video traffic (Sankar and Chellamuthu, 2009). In this study, the application of simulation environment for analyzing the effect of increased video traffic on the responsive flows is considered. The simulation environment for video traffic in Linux platform along with SFQ and SFB algorithms are described.

SCHEDULING ALGORITHMS

Simulation environment: The video simulation environment developed by researchers is shown in Fig. 1. It consists of ns2, ffmpeg codec and its service utilities (Sankar and Chellamuthu, 2010). The ns2 is an object oriented network simulator.

It supports simulation of TCP, UDP, media access control layer protocols and various routing and multicast protocols over both wired and wireless networks. There are four schedulers available in the network simulator: linked list, heap, calendar queue and real time scheduler.

The scheduler initiates bunch of events one after other and runs it into completion. The software tool ffmpeg in the MPEG4 format is chosen as video codec for encoding the test video because it is

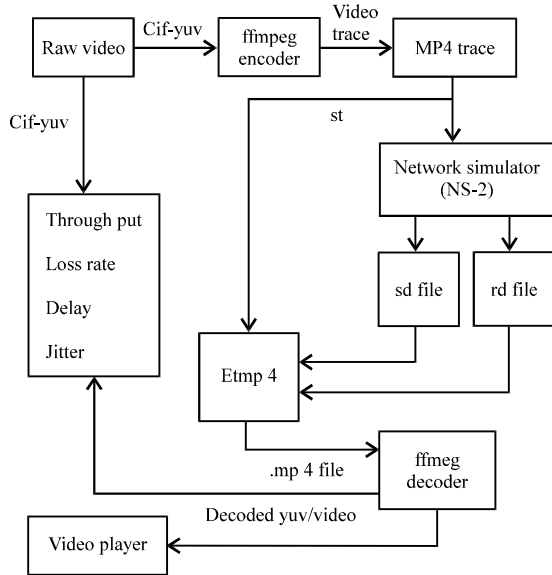


Fig. 1: Simulation environment for video traffic

widely used and provides better documentation for comparing video qualities. The service utilities of video codec consists of error generator (eg), etmp4, psnr, hist and mos. The ns2 simulator has several tools such as graphical animator (nam), xgraph and gnuplot. These utilities help in generating and visualizing the simulation results.

Stochastic fair queue algorithm: Fair Queuing (FQ) is a queue scheduling discipline which ensures that each flow has a fair access to network resources. It also prevents a bursty flow from consuming more than its fair share of the output port bandwidth.

In FQ, packets are first classified into flows by the system and then assigned to a queue that is specifically dedicated to that flow. The queues are then serviced one packet at a time in round-robin order. The empty queues are skipped. The key to classify a flow is based on source address, source port and destination address. It is not practical to have one queue for each flow (Demers *et al.*, 1989).

The SFQ is a modification to FQ. It employs a hashing algorithm which divides the traffic over a limited number of queues. It is called Stochastic because it does not really allocate a queue for each flow. Because of the hash, multiple flows might end up in the same bucket, reducing the effective speed of the flow.

To prevent this situation, the SFQ changes its hashing algorithm quite often so that any two colliding flows will only do so for a short time. The limitation of this algorithm is that the output queue should always be full

and it will not rate limit the non-responsive flows. This drawback is overcome by the SFB algorithm which is discussed here.

Stochastic fair blue algorithm: SFB combines BLUE algorithm and Bloom filters to enforce fairness among the flows using an extremely small amount of state and large buffer space. The SFB algorithm identifies and rate-limits non-responsive flows based on accounting mechanisms (Feng *et al.*, 2002). The SFB maintains $N \times L$ accounting bins. The bins are organized in L levels with N bins in each level. SFB also maintains independent hash functions, each associated with one level of the accounting bins. The pseudo code of hashing function is shown here.

```

On every packet arrival:
Calculate hashes  $h_0, h_1, \dots, h_{L-1}$ 
Update bins at each level
for  $i = 0$  to  $L-1$  do
  if  $(B[i][h_i].qlen > b \text{ bin size})$  then
     $B[i][h_i].pm += \text{delta}$ 
Drop packet:
  else if  $(B[i][h_i].qlen == 0)$  then
     $B[i][h_i].pm -= \text{delta}$ 
  end if
end for
 $pmin = \min(B[0][h_0].pm, B[1][h_1].pm, \dots, B[L-1][h_{L-1}].pm)$ 
if  $(pmin == 1)$  then
  ratelimit()
lse
  Mark or drop packet with probability  $pmin$ ;
end if
On every packet departure:
Calculate hashes  $h_0, h_1, \dots, h_{L-1}$ 
Update bins at each level:
for  $i = 0$  to  $L-1$  do
  if  $(B[i][h_i].qlen == 0)$  then
     $B[i][h_i].pm$ 
  end if
end for
  
```

The typical parameters of SFB algorithm are Q_{len} , Bin-size, freetime, N , L , Boxtime and Hinterval. The Bin-size The buffer capacity of each bin is equal to the Bin-size. The Q_{len} is the queue length of each bin. Each hash function maps a flow into one of the N accounting bins in that level. The accounting bins are used to keep track of queue occupancy statistics of packets belonging to a particular bin. Each bin in SFB keeps a marking/dropping probability P_m which is updated based on bin occupancy. As a packet arrives at the queue, it is hashed into one of the N bins in each of the L levels. If the number of packets mapped to a bin goes above a certain threshold (i.e., the size of the bin), P_m for the bin is increased. If the number of packets drops to zero, P_m is decreased. In SFB non-responsive flow quickly drives P_m to 1 in all of the L bins which are hashed. The responsive

flows may share one or two bins with non-responsive flows. If the number of non-responsive flows is extremely large compared to the number of bins, a responsive flow is likely to be hashed into at least one bin that is not polluted and has a normal Pm value. The decision to mark a packet is based on Pmin. If Pmin is 1, the packet belongs to a non-responsive flow and it is rate-limited.

SIMULATION AND RESULTS

Two raw cif-yuv video files with resolution 352×288, 30 frames sec⁻¹, 8 sec play time and a bit rate of 47.3 kbps is encoded using MPEG4 to generate video trace file of 250 and 300 frames, respectively. A trace file st is created using the mp4 trace utility for the video file and it is attached to a tcl program which simulate the given network topology (Fall and Varadhan, 2005). When the tcl file is made to run, it converts the st file to a video data file and writes it with dat extension. It also generates a sender dump (sd) and a receiver dump (rd) files. Using etmp4 tool along with sd, rd and st files the video file is generated and saved with .mp4 extension. This .mp4 is converted to yuv using the ffmpeg that can be played on the players like Video LAN Client (VLC) for visualization.

The model of the network used for simulation is shown in Fig. 2. The simulation is carried out for two different scenarios. For both scenarios 1 Mbps and 2 Mbps bandwidths are applied in the bottleneck link between n5 and n6.

In the scenario I, Non-Responsive Flow (NRF) is 20% and Responsive Flows (RF) are 80% whereas in scenario II, NRF is 40 and RF is 60%. The flows are traversing from both sides of the bottleneck link.

The network comprises of sources n0, n1, n2, n11, n12 and n3, n4, n8, n9, n10 as destinations with three routers n5, n6 and n7. The flows that are considered in the simulation are from n0-n10 and n1-n8, respectively. The parameters used for simulation are shown in the appendix.

Packet loss: The variation of packet loss of NRF and RF flows for SFB and SFQ algorithms at the bottleneck over a period of simulation is shown in Fig. 3. It shows that the RF flow has low packet loss. The packet losses for responsive and non-responsive flows for different bandwidths are shown in Table 1.

It is seen from the Table 1 that the percentage of packet loss for responsive flows is reduced when non-responsive flows are doubled. Further it is observed that SFB algorithm penalizes the non-responsive flow by dropping more packets so that responsive flow is not affected. It is seen that both algorithms are effective in maintaining the correct flow ratio in the link when NRF is increased.

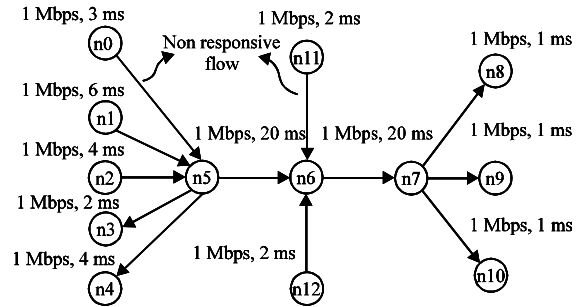


Fig. 2: Model of the network for simulation

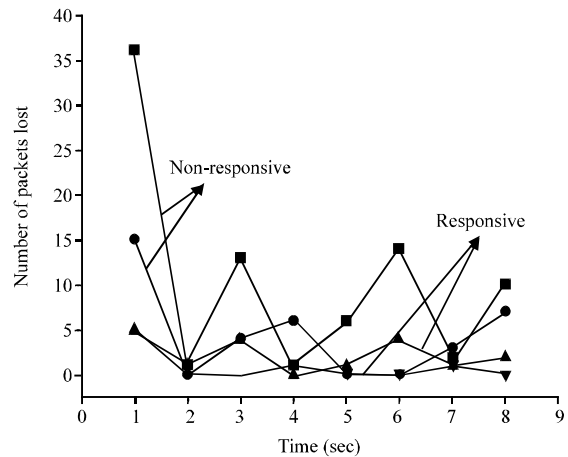


Fig. 3: Packet loss for scenario II with 1 Mbps bandwidth

Table 1: Percentage packet loss at 1 Mbps and 2 Mbps bandwidths

Algorithm	Scenario I		Scenario II	
	NRF	RF	NRF	RF
1MB				
SFB	19.3	9.30	21.27	7.29
SFQ	7.9	2.70	7.39	2.12
2MB				
SFB	16.77	1.70	16.69	1.28
SFQ	4.50	0.59	4.50	0.44

Table 2: Percentage throughput at 1 Mbps and 2 Mbps bandwidth

Algorithm	Scenario I		Scenario II	
	NRF	RF	NRF	RF
1MB				
SFB	79.9	90.6	77.9	89.4
SFQ	89.5	94.5	89.6	95.4
2MB				
SFB	82.4	96.7	82.5	96.6
SFQ	94.6	98.8	94.6	98.9

Throughput: The throughputs of the responsive and non-responsive flows are shown in Table 2. It clearly indicates that throughput of responsive flow is not deteriorated when the non-responsive flow is increased from 20-40%. The SFB algorithm could effectively detect the video (non-responsive flows) and rate-limit them. It is not adaptive because of the fixed box-time used in the simulation.

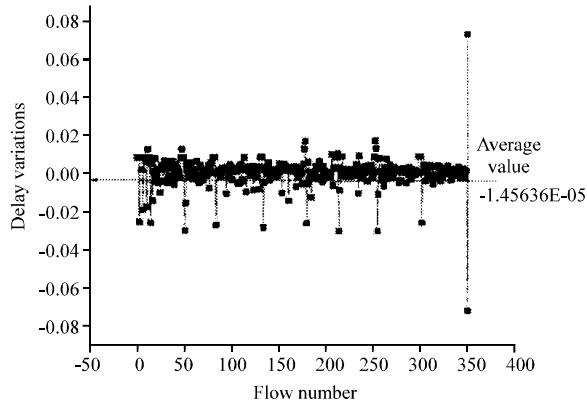


Fig. 4: Jitter of SFB algorithm for scenario I at 2 Mbps

Table 3: Jitter for scenario I

Parameters (1 Mbps)	SFB		SFQ	
	RF	NRF	RF	NRF
Mean (μ sec)	9.52000	7.28000	10.39000	11.32000
Variance (%)	1.08000	1.00000	3.81000	2.68000
Shaping parameter (α)	0.27435	0.92035	0.62839	0.99469
Scaling factor (β)	0.00568	0.00683	0.00760	0.01104

Jitter: High jitter degrades the quality of the video in multimedia applications. The delay variations of NRF for a worst case scenario over a period of simulation are shown in Fig. 4. It is observed from the plot that the jitter is highly fluctuating and difficult to extract information for quality assessment. The Weibull distribution is therefore considered for the analysis of jitter.

The analysis in terms of mean, variance, shaping parameter and scaling factor is carried out and is shown in Table 3. The Weibull distribution relating jitter and its duration of occurrence expressed as a percentage over a period of simulation is shown in Fig. 5 and 6, respectively. The distribution plot clearly indicates that as the value of α decreases from 0.92-0.27, the distribution curve falls down rapidly. It is inferred that the jitter decreases with decrease in α . The Jitter is high for video traffic (NRF) when compared to RF. From Table 3, it is observed that SFB algorithm gives less value of mean and variance for jitter when compared to values obtained for SFQ. It is found that when the bandwidth of the link is increased the mean value of jitter decreases with variance remaining nearly constant. Therefore, the SFB performs better than SFQ algorithm when video traffic is modeled as NRF.

Link utilization: The incoming and outgoing traffic in the duplex link has been calculated for both responsive and non responsive flows. The percentage of link utilization of the responsive flow is 71% and that of the video flow is 29%. This clearly indicates that the link was fairly utilized as the incoming traffics attached to the network was 20 UDP and 80% TCP.

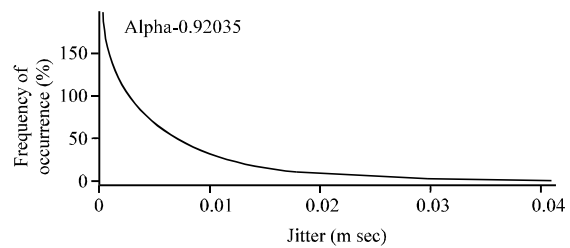


Fig. 5: Jitter of SFB algorithm for scenario I at 2 Mbps

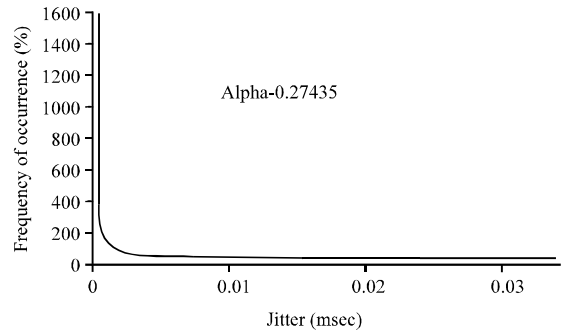


Fig. 6: Jitter distribution for responsive flows

CONCLUSION

In this study, the simulation environment developed by the researchers is used to study the performance analysis of responsive and non-responsive flows using SFB and SFQ queue management algorithms. The video /UDP traffic was modeled as non-responsive flow instead of UDP/CBR traffic as the usage of multimedia application is steadily increasing in the network. The simulation was carried out for two different scenarios with 1 Mbps and 2 Mbps bandwidth at the bottleneck link. The results show that both algorithms protect the RF in the presence of increased NRF because the link bandwidth was fairly utilized.

ACKNOWLEDGMENTS

The researchers wish to acknowledge the technical support given by Mr. Anand and Mr. Raj manickkam of Kumarn system for this research.

APPENDIX

Parameters	Specifications
Number of nodes	12 nodes
Link parameters	Duplex/2 Mband 1Mb
	Bandwidth/20 msec delay
Packet sizes	Video-1052 max/TCP 1040 max
Link analyzed	5 and 6
Buffer size	50 packets
Cross flows	Two
SFB Pbox time	1 m sec
SFQ buckets	16
Metrics used	Throughput and link utilization

REFERENCES

- Cobb, J.A., M.G. Gouda and A. El-Nahas, 1998. Time-shift scheduling-fair scheduling of flows in high-speed networks. *IEEE/ACM Trans. Network.*, 6: 274-285.
- Demers, A., S. Keshav and S. Shenker, 1989. Analysis and simulation of a fair queueing algorithm. *Proceedings of the ACM Symposium on Communications Architectures and Protocols*, Sept. 19-22, ACM, Austin, TX, USA., pp: 1-12.
- Fall, K. and K. Varadhan, 2005. The NS manual. VINT Project, May 17, 2005.
- Feng, W., D. Kandlur, D. Saha and K. Shin, 2002. Blue: A new class of active queue management algorithms. *IEEE/ACM Trans. Network.*, 10: 513-528.
- Kim, J.H., H. Yoon and I. Yeom, 2010. Active queue management for flow fairness and queue stability. *IEEE Trans. Parallel Distributed Syst.*, 13: 998-1000.
- Li, L. and G. Lee, 2003. Adaptive queue management with restraint on non-responsive flows. *J. Systemics Cybernetics Informatics*, 1: 40-45.
- Sankar, P. and C. Chellamuthu, 2009. Study and simulation of virtual clock scheduling algorithm for video streaming. *Int. J. Comput. Theory Eng.*, 1: 509-514.
- Sankar, P. and C. Chellamuthu, 2010. Study and simulation of video streaming. *Int. J. Comput. Appl.*, Vol. 32, No. 4.
- Zhao, Z., S. Darbha and A.L.N. Reddy, 2004. A method for estimating the proportion of nonresponsive traffic at a router. *IEEE/ACM Trans. Network.*, 12: 708-718.