

Modeling and Analysis of Adaptive Buffer Sharing Scheme for Consecutive Packet Loss Reduction in Broadband Networks

Sakshi Kausha, R.K Sharma

Abstract—High speed networks provide realtime variable bit rate service with diversified traffic flow characteristics and quality requirements. The variable bit rate traffic has stringent delay and packet loss requirements. The burstiness of the correlated traffic makes dynamic buffer management highly desirable to satisfy the Quality of Service (QoS) requirements. This paper presents an algorithm for optimization of adaptive buffer allocation scheme for traffic based on loss of consecutive packets in data-stream and buffer occupancy level. Buffer is designed to allow the input traffic to be partitioned into different priority classes and based on the input traffic behavior it controls the threshold dynamically. This algorithm allows input packets to enter into buffer if its occupancy level is less than the threshold value for priority of that packet. The threshold is dynamically varied in runtime based on packet loss behavior. The simulation is run for two priority classes of the input traffic – realtime and non-realtime classes. The simulation results show that Adaptive Partial Buffer Sharing (ADPBS) has better performance than Static Partial Buffer Sharing (SPBS) and First In First Out (FIFO) queue under the same traffic conditions.

Keywords—Buffer Management, Consecutive packet loss, Quality-of-Service, Priority based packet discarding, partial buffer sharing.

I. INTRODUCTION

THE QoS requirements are generally specified in terms of packet loss and transmission delay. Assuming Internet will continue to become congested due to scarcity of resources such as bandwidth and buffer, this proposition leads to several possible approaches for better QoS [1][2]. In packet-switched networks, congestion due to statistical traffic fluctuation can result in packet loss. There are several different mechanisms to avoid congestion: admission control, rate-based control, and resource reservation. In this context, congestion control through adequate buffering is becoming particularly significant to minimize the probability of packet-loss and packet delay.

Buffer management is a fundamental technology to provide

Manuscript received April 23, 2007.

Sakshi Kaushal is with Department of Computer Science and Engineering, University Institute of Engineering and Technology, Panjab University, Chandigarh, India (phone: +919872236600; emailsakshi2k1@rediffmail.com and Sakshi@pu.ac.in).

R.K Sharma is with School of Mathematics and Computer Applications, Thapar University, Patiala, India (phone: +91-175 2393378 ; e-mail: rksharma@tiet.ac.in).

QoS control mechanisms, which control the assignment of buffer resources among different flows and flow aggregations according to certain policies. This is because, unlike the deterministic multiplexing where each connection is allocated its peak bandwidth, statistical multiplexing allows several connections which might be very bursty at times, to share the same link based on their traffic characteristics in the hope that statistically they will not all burst, at the same time. However, congestion still happens when multiple packets blasting away at the peak rate simultaneously through different incoming links attempt to reach the same outgoing link during the same packet slot time. In this case, only one packet is allowed to go through the network while the others must be stored in buffers. At this time, a switch buffering strategy as well as the buffer size becomes important because buffers are required to secure low packet-loss rate by providing a place to guard against packet-loss when the switch is overloaded with bursty traffic. The choice of either of them can have a dramatic effect on the performance of the switch. If packet-loss is experienced due to overflowing buffers, this will introduce degradation in the overall system performance.

It is necessary to design a network that can dynamically adapt to the changing needs of QoS of the clients, but still provide guarantees per session. Some applications such as video and voice have stringent timing requirements but can tolerate some degree of packet loss. On the other hand, applications like file transfer have low loss tolerance. But large delay is not a problem. The main function of packet discarding mechanisms in congested network is to control consecutive packet loss of different packet discarding algorithms. The control schemes usually categorize and manage the packets loss priorities assigned to them. Priority-based discarding has been shown to have the potential of improving system performance for voice and video traffic. An efficient buffer management strategy maintains a high throughput under a variety of load conditions, while meeting the packet loss requirements of various traffic classes. Early work on buffer management focused on the problem of sharing a finite storage space among several output queues for traffic of a single loss priority [3].

Buffer management schemes can be categorized into three classes: complete partition policy based, complete sharing policy based, and PBS (Partial Buffer Sharing) policy based.

In Complete partition, buffer space is statically partitioned into different queues each for a single class, which can use simple control mechanisms to achieve precise differentiated performance between classes. However, complete buffer partitioning reduces the buffer utilization heavily and increases the overall packet loss rate because arrival packets may be discarded even when there are buffer resources available.

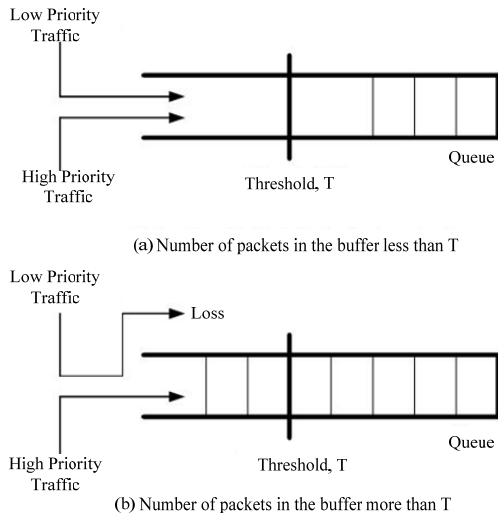


Fig. 1 SPBS Scheme for two priorities incoming traffic

To increase the overall system buffer utilization, dynamic-partition based schemes have been proposed [4][5][6].

Complete buffer sharing policy can achieve highest buffer utilization because all of the buffers are always occupied unless there is not enough arriving traffic. Traditional Internet use FIFO with drop-tail mechanism and achieve high buffer utilization, but does not support service differentiation and fairness control. Push-out scheme is an enhancement of FIFO and drop-tail, which supports multiple classes of traffic. In Push-out, the arriving packets are allowed to enter the buffer as long as there is space, and when the buffer fills up, an incoming packet is allowed to enter by selectively overwriting another packet that is already in the buffer that is of lowest priority [7]. Multi-Queue based Push-out policy can achieve highest buffer sharing as well as service differentiation and fairness assurance [8]. [8] presents a PLR (Proportional Loss Rate) dropper to support proportional differentiated services using multi-class push-out policy. But large computing complexity is its inevitable disadvantage as other push-out schemes, especially with the number of service classes increasing.

The PBS scheme controls incoming traffic from different priority classes based on threshold in buffer. When the buffer level is below threshold, PBS accepts both high priority and low priority packets and when the buffer level is over a predetermined threshold, low priority packets cannot access the buffer and are discarded. In other words, high priority

packets continue to access the buffer unless it is full. The schematic diagram for this mechanism of priority based traffic control is represented in Figure 1. The low priority traffic is allowed to enter into buffer only if the buffer occupancy level is less than the threshold, T . For high priority traffic the complete buffer is accessible irrespective of the buffer occupancy level and threshold value. In partial buffer sharing schemes, the threshold is considered to be constant, which is referred to as Static Partial Buffer Sharing (SPBS) schemes [9][10]. However, we will see in Section IV that its performance suffers from the fact that it is not dynamic. The challenge in designing a SPBS scheme is to select optimal threshold value to obtain desired relative packet loss ratio among the two classes of traffic.

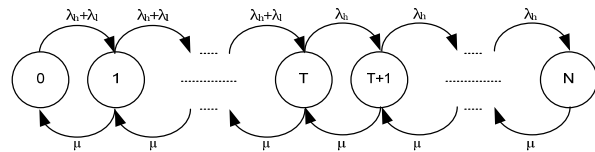


Fig. 2 State Transition Diagram for Static Partial Buffer Sharing Scheme

Figure 2 shows the state transition diagram of buffer under the PBS mechanism. The traffic model has two input streams for different types of traffic, i.e., high priority and low priority traffic. The packets arrive at system according to Poisson process with arrival rates λ_h and λ_l for high priority and low priority traffic respectively. Both types of packets are stored at a common buffer which has fixed service time denoted by μ for both traffics. The capacity of buffer is N and threshold level is T . According to PBS mechanism, low priority packets are admitted to buffer only when buffer occupancy is less than T ($< N$) while the buffer access of high priority traffic is limited by its full capacity.

In high speed networks, the two major performance measures are the end-to-end transfer delay and end-to-end packet loss probability. The quality of traffic including video, voice and other data signals is sensitive to consecutive packet losses rather than single packet loss. Therefore, the proper performance measure for the traffic in such networks is consecutive packet loss probabilities. The goal is to accommodate more incoming packets from various sources and smooth out the burst arrival rate while limiting the overhead of the switch within a predefined size. Therefore, a novel scheme of adaptive threshold to fairly regulate the sharing of memory among queue for traffic of loss priority is proposed. This model can be applicable to any number of packet classes. The adaptive threshold scheme which is an improvement of SPBS scheme, adapts to changes in traffic conditions. Whenever the load changes, the system will go through a transient and guarantee the packet loss ratio performance between classes and improves the buffer utilization as possible.

In past [11], some non-parameter schemes have been

proposed based on to fuzzy theory. These schemes are focused on providing flexible buffer management to ensure smooth packet loss behavior, but their disadvantages are too complex and the trade-off between performance and complexity is not very good.

The rest of the paper is organized as follows: the proposed adaptive partial buffer sharing scheme is presented in section II. Section III describes the performance analysis of ADPBS scheme. In first part of this section, analytical model used to control loss probability ratios with combination of control parameters is discussed and proposed algorithm is in the second part. Simulation results showing the comparative performance of the proposed ADPBS, SPBS and FIFO is given in Section IV. Finally, section V concludes the paper.

II. ADPBS SCHEME

ADPBS scheme is an improvement upon the SPBS scheme. The packet discarding decision is according to the relation between current queue occupancy and discarding thresholds and also it can be generalized to support multiple discarding priorities. In this section we shall present the detailed modeling and analysis of ADPBS scheme.

Assume that the queuing system can hold up to N packets and arrival packets can be classified to D discarding priorities. A series of discarding thresholds $T(d)$, where $1 \leq d \leq D$, would be deployed. This will satisfy the following rule:

$$0 \leq T(1) \leq T(2) \leq \dots \leq T(D-1) \leq T(D) \quad (1)$$

In this way, the buffer space is virtually partitioned into the different segments based on priority level of the traffic and is controlled by the threshold value. An arriving packet with priority d will be admitted into the queue only if current queue occupancy is less than $T(d)$; otherwise it will be discarded. Once a packet is admitted, it is never subject to discard.

A. ADPBS Scheme for two priorities

Assuming a finite buffer of size N and source traffic with two classes of priorities, there can be at most two classes of packets, say class 1 with low priority and class 2 with high priority. Also, there will be just one discarding threshold T involved as illustrated in Figure 3. Any arriving packet can be admitted into the queue if the current queue occupancy is less than the discarding threshold T , otherwise only high priority packets can be admitted if queue is not full and low priority packets will be discarded while entering. For each class of packets, a counter is maintained to count the number of lost packets with initial value set to zero – C_h for high priority and C_l for low priority. Each counter is assigned two parameters to control the threshold – loss bound and modification step level. The loss bound is limit for the counter to invoke change in threshold. The loss counter C_h when reaches its loss bound limit b_h , the threshold T is made to decrease by modification step level t_h . Similarly when the loss counter C_l reaches the loss bound limit b_l , the threshold T is made to increase by

modification step level t_l . The amount of limit movement by parameters t_h and t_l are based on the bursting of the traffic, the more burst the traffic, the large the values should be set to reduce excessive fluctuations. The frequency of threshold movement is determined by the size of b_h and b_l . When the values of b_h and b_l are set to be very large, the threshold is not changed very often and ADPBS scheme generates to the SPBS scheme.

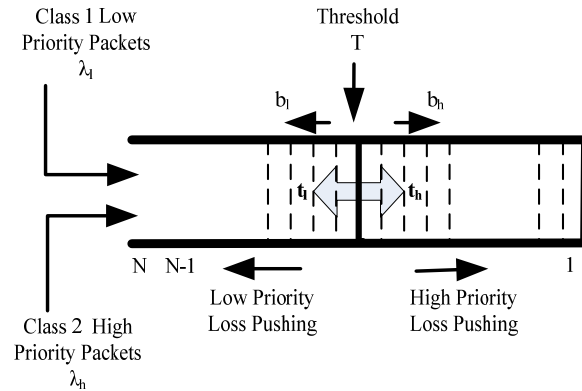


Fig. 3 ADPBS queuing model for two priorities

The scheme is based on the assumption that in a short time slot, incoming traffic tends to keep steady, especially with self similar traffic pattern. If a number of low priority packets are lost, there is probability that heavy low priority traffic will arrive in a short while and the discarding threshold should be increased to provide sufficient buffer space for them. If lots of high priority packets are lost, the discarding threshold should be decreased to provide sufficient buffer space for high priority packets correspondingly.

B. ADPBS Scheme for multiple priorities

The most difference to two-priority case is that, there are two thresholds related to each discarding priority, except the lowest and highest priority one. One give the usable buffer space up-bound for it, and other is sharing space with lower priority, we call them upper thresholds and lower thresholds separately.

Each discarding priority maintains one loss counter for each relative threshold and pushes the threshold to move outward when loss counter exceeds the loss up-bound parameter as same as two-priority case. If for a certain priority too much packets were discarded, ADPBS scheme will increase its upper thresholds to make more buffer space, while decrease its lower thresholds to reduce the sharing space simultaneously. During a long period, the relative loss ratios incline to keep approximately stable to some degree.

III. PERFORMANCE ANALYSIS

A. Analytical Model

For analyzing the performance of the proposed algorithm, two priority input traffic is considered to maintain the

complexity level low. With two priority traffic, there are packets with high priority and low priority. The buffer is the common resource for both high priority and low priority traffic. The threshold controls the access to buffer for the incoming traffic – allowing both types of traffic if the buffer occupancy is below threshold level and if the occupancy level is more than threshold it accepts only high priority incoming traffic. The total arrival rate for high priority is represented as λ_h and that for low priority as λ_l . The service rate of the queue is represented as μ . The traffic load for the queue is given by $\rho = (\lambda_h + \lambda_l) / \mu$.

Equation (2) shows P_l is the packet loss ratio of low priority packets. The rate of threshold movement towards the left (higher positions) is proportional to low priority packet loss rate and the values of b_l and t_l , and is given by:

$$P_l \lambda_l \left(\frac{t_l}{b_l} \right) \tag{2}$$

Similarly, in (3), P_h is the packet loss ratio of high priority packets. The rate of threshold movement towards the right (lower positions) in the buffer is associated with the high priority packet loss rate and the values of b_h and t_h , and is given by:

$$P_h \lambda_h \left(\frac{t_h}{b_h} \right) \tag{3}$$

If the threshold moves towards the right, it yields enough space only for high priority packets to be accommodated in buffer at the cost of loss of low priority packets due to reduced buffer space for those. Hence, loss probability of low priority packets increases. Similarly, the threshold movement towards left results in increased loss probability of high priority packets. This tradeoff results in equilibrium of threshold when two equations are equal as shown in (4).

$$P_l \lambda_l \left(\frac{t_l}{b_l} \right) = P_h \lambda_h \left(\frac{t_h}{b_h} \right) \tag{4}$$

$$\left(\frac{P_h}{P_l} \right) = \left(\frac{\lambda_l}{\lambda_h} \right) \left(\frac{t_l}{t_h} \right) \left(\frac{b_h}{b_l} \right) \tag{5}$$

Equation (5) clearly indicates that the ratio of loss probability high priority and low priority packets depend on input traffic arrival rate and threshold control parameters. The minimum value for the ratio of P_h and P_l is when the threshold reaches first position (position 1 of buffer) where the low priority packets are discarded on arrival. At this threshold level, the value of this ratio becomes P_h . When the threshold level reaches the last position (position $N-1$ of buffer), the ratio P_h and P_l becomes 1, since both low priority and high priority packets get access to complete buffer.

Considering the minimum and maximum values for the ratio of P_h and P_l , its range of values may be expressed as:-

$$P_h \leq \left(\frac{P_h}{P_l} = \frac{\lambda_l t_l b_h}{\lambda_h t_h b_l} \right) \leq 1 \tag{6}$$

ADPBS scheme for multiple priorities can be analyzed in the same way and the following relation holds:

$$P_{d+1} \leq \left(\frac{P_{d+1}}{P_d} = \frac{\lambda_d t_d b_{d+1}}{\lambda_{d+1} t_{d+1} b_d} \right) \leq 1 \tag{7}$$

Equation (6) and (7) allow us to select the control parameter value to get the expected relative loss ratios, if the arriving traffic pattern has been clear. This is a very important characteristic of ADPBS scheme, for it has solved the thresholds setting problem for partial buffer sharing scheme.

With ADPBS scheme, all the packets of different priority share the same queue space, and compete for shared buffer resources one another. Consequent thresholds adjustment produces a negative feedback effect, which can adjust threshold to optimal value quickly when traffic load changes.

TABLE I
VARIABLES USED IN THE ALGORITHM

Variable	Description of the variable used in Algorithm
total_pkt_in_sys	Counter for Total Number of packets used in Simulation
prio	Priority value (High / Low) for the arrived packet
priority	Function to determine priority
qlen	Queue size
qlim	Buffer size
prev_counter_hi_prio	Counter for High Priority packet to check consecutive packet loss
prev_counter_low_prio	Counter for Low Priority packet to check consecutive packet loss
loss_high	Counter to register Low priority packet loss
loss_low	Counter to register High priority packet loss
loss_bound_hi	Limit for high priority packet loss counter when reached decreases threshold
loss_bound_low	Limit for low priority packet loss counter when reached increases threshold
TH	Threshold value of the buffer

Therefore, the variation of packet loss ratios is small with ADPBS scheme, as well as the relative loss ratios tend to be stable to some degree. The experimental results in section IV will show these advantages of ADPBS scheme more clearly.

B. Proposed Algorithm

The description of the variables used in the algorithm is given below in Table I. The proposed algorithm is given below:

```

for every arrival packet pkt
  total_pkt_in_sys ++
  /*determine priority of arrived packet*/
  prio = priority(pkt)
  if qlen < qlim
  /*there is enough buffer space for packet*/
  accept packet
  else
  discard packet

  /*for high priority packet*/
  if pkt has high priority
  /*for consecutive packet loss*/
  if (prev_counter_hi_prio == total_pkt_in_sys - 1)
  loss_high ++
  prev_counter_hi_prio = total_pkt_in_sys
  if (loss_high == loss_bound_high)
  loss_high=0 /*reset packet loss counter*/
  if threshold can be decreased further
  TH = TH -1
  end
  end
  else
  prev_counter_hi_prio = total_pkt_in_sys
  loss_high +1
  end

  /*for low priority packet*/
  else
  if (prev_counter_low_prio == total_pkt_in_sys - 1)
  loss_low ++
  prev_counter_low_prio = total_pkt_in_sys
  if (loss_low == loss_bound_low)
  loss_low = 0 /*reset packet loss counter*/
  if threshold can be increased further
  TH = TH + 1
  end
  end
  else
  prev_counter_low_prio = total_pkt_in_sys
  loss_low = 1
  end
  end
end
endfor

```

IV. SIMULATION RESULTS

In this paper, we have illustrated that the ADPBS attempts

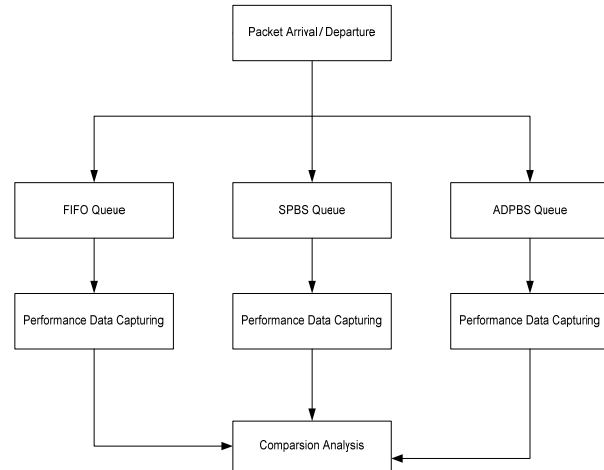


Fig. 4 Simulation model for comparing and analyzing the results for different queues

to reduce consecutive packet loss as compared with SPBS and FIFO. Using the above analysis, we can control the loss of consecutive high priority packets and low priority packets through the combinations of the parameters like $-\lambda_h, \lambda_l, t_h, t_l, b_h$ and b_l . In this section, we provide our simulation results to illustrate the performance of ADPBS, SPBS and FIFO queues.

The model given in Figure 4 represents that the simulation results are obtained three queues under the same traffic conditions. The common values used in the simulation are buffer size, $N=20$ and service rate, $\mu=20$. The relative packet loss behavior is studied by deploying exponential distributed on/off model to generate packets traffic common for the three different queue mechanisms. For analyzing the performance, we have divided the simulation in three different parts, as discussed below:

A. Load Factor Variation

In this section, the performance of adaptive threshold queue is discussed and compared with static threshold queue and FIFO queue.

Table II gives the results of the simulation run with the proposed algorithm. The following values are assigned to the

TABLE II
TOTAL CONSECUTIVE PACKETS LOSS OF HIGH AND LOW PRIORITY PACKETS
FOR DIFFERENT LOAD RATIOS

Load	FIFO		SPBS		ADPBS	
	High Priority	Low Priority	High Priority	Low Priority	High Priority	Low Priority
0.7	60	50	23	52	20	99
0.75	36	70	16	60	16	60
0.8	173	110	60	80	34	130
0.85	274	250	126	297	99	327
0.9	455	409	209	589	165	540
0.95	519	498	275	664	239	687

variables: $t_h=2$, $t_l=2$, $b_h=2$ and $b_l=6$. The values of λ_l and λ_h are varied to change the load conditions keeping the service rate, $\mu=20$ as constant. This table captures the number of consecutive packets lost (high priority and low priority both) and tabulates the comparative values FIFO, SPBS and ADPBS with reference to a particular load value.

Figure 5 gives consecutive high priority packets lost in FIFO Queue, Static Threshold Queue and Adaptive Threshold Queue for different load ratios. The results show that the curve for adaptive threshold is always lower among the three curves. For higher values of the load ratio, the performance of adaptive threshold queue is better, where as for its lower values there is not significant difference of performance among three curves. At load ratio of 0.7, the loss ratio of static and adaptive is 1.15; for FIFO and ADPBS this ratio is 3 where as this ratio increases to 1.76 and 5.09 respectively for load ratio 0.8.

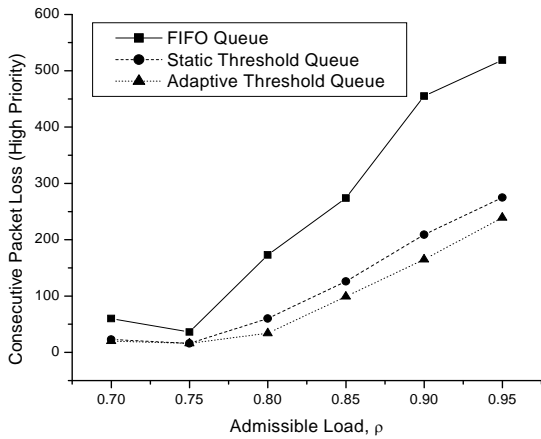


Fig. 5 Consecutive packet loss of high priority packets for different load ratios

Figure 6 gives the low priority packets loss for different load ratios. This again illustrates that for lower load ratios the difference in performance is insignificant, however when the load ratio is higher, the FIFO queue performance goes better.

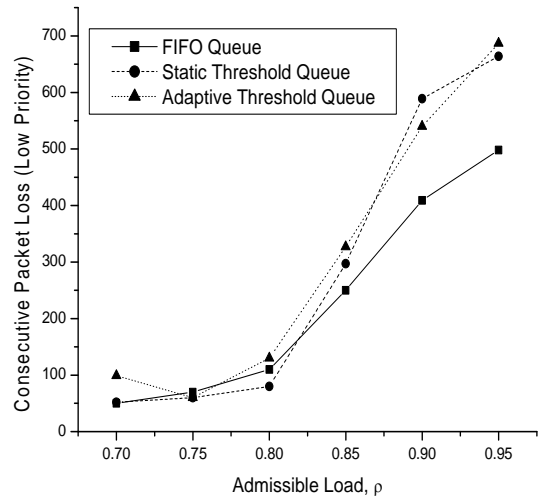


Fig. 6 Consecutive packet loss of low priority packets for different load ratios

The main fact for this is: as on higher values of load ratios, more of the buffer space gets allocated for accommodating the high priority packets, so low priority packets suffer loss. At load ratio of 0.7, the loss ratio of static and adaptive is 0.53 while it increases to 1.09 when load ratio is 0.9.

B. Input traffic mix variation

In this simulation, the impact of the high priority packets and low priority packets traffic mix is studied. Table III gives the results observed after the simulation run with the above given algorithm. The following values are assigned to the variables: $t_h=2$, $t_l=2$, $b_h=2$ and $b_l=6$. The values of λ_l and λ_h are varied to change the input traffic mix for the same load conditions, that is, keeping the total arrival rate, λ and service rate, $\mu=20$ as constant. This table captures the number of consecutive packets lost (high priority and low priority both) and tabulates the comparative values for FIFO, SPBS and ADPBS queues with reference to a particular value of input traffic mix.

TABLE III
TOTAL CONSECUTIVE PACKETS LOSS OF HIGH AND LOW PRIORITY PACKETS FOR DIFFERENT INPUT TRAFFIC RATIOS

High Priority Packet Arrival Rate, λ_h	Low Priority Packet Arrival Rate, λ_l	Input Traffic Ratio (λ_h / λ_l)	FIFO		SPBS		ADPBS	
			High Priority	Low Priority	High Priority	Low Priority	High Priority	Low Priority
2	12	0.17	43	124	0	120	12	58
3	11	0.27	66	200	0	209	24	92
4	10	0.40	46	147	0	166	12	107
5	9	0.56	79	135	0	220	31	134
6	8	0.75	10	29	0	46	0	47
7	7	1.00	60	50	23	52	20	99
8	6	1.33	95	61	27	95	24	142
9	5	1.80	71	48	28	57	18	75
10	4	2.50	76	23	44	31	26	31
11	3	3.67	77	21	51	23	35	23
12	2	6.00	65	21	36	21	26	21

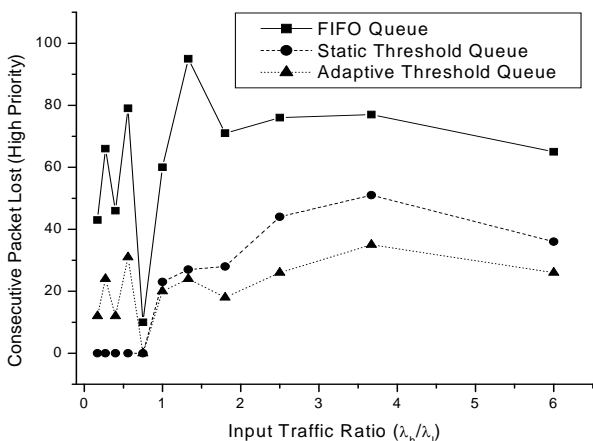


Fig. 7 Consecutive packet loss of high priority packets for different input traffic ratios

Figure 7 gives the consecutive packets lost for the ratio of high priority input traffic. The results are again compared with FIFO queue and static threshold queue to evaluate the overall performance of ADPBS. Below the input traffic ratio of 0.75, the static queue shows good results with minimum loss of the consecutive high priority packets. However, adaptive queue starts performing better beyond input traffic ratio of 0.75. At the input traffic ratio of 0.17, the consecutive packet loss ratio of high priority packets in FIFO and adaptive queue is 3.58; for static and adaptive this ratio is 0 and at input traffic ratio of 1.80, the loss ratios become 3.94 and 1.56.

The results for loss of consecutive low priority packets are shown in Figure 8 for various input traffic ratios. It is illustrated here that below input traffic ratio of 0.75, the

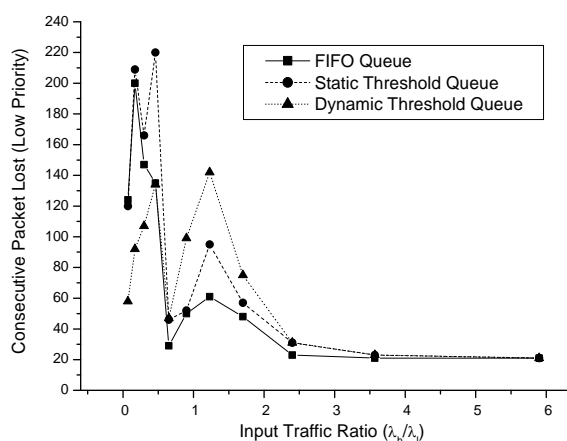


Fig. 8 Consecutive packet loss of low priority packets for different input traffic ratios

adaptive threshold queue performs better. However, its performance deteriorates significantly beyond this point. At input traffic ratio of 0.17, the consecutive packet loss ratio of low priority packets for FIFO and adaptive queue is 2.14 whereas for static and adaptive queue, this ratio is 2.07 and at input traffic ratio of 1.33, the loss ratios become 0.43 and 0.67.

It is also observed that the dynamic queue performs better for the kind of traffic which has higher proportion in the input traffic mix. For example, when the input traffic mix has major content of high priority packets, the consecutive packet loss of high priority packets significantly decreases as compared to low priority packets and vice-versa. This characteristic of ADPBS illustrates its efficient control and adaptive nature.

TABLE IV
CONSECUTIVE PACKETS LOSS RATIOS OF HIGH AND LOW PRIORITY PACKETS FOR DIFFERENT LOSS BOUND RATIOS

Modification Step		Modification Step Ratio (t _h /t _l)	Loss Bound		Loss Bound Ratio (b _h /b _l)	Consecutive Packet Loss		Consecutive Packet Loss Ratio
High Priority, t _h	Low Priority, t _l		High Priority, b _h	Low Priority, b _l		High Priority	Low Priority	
1	2	0.5	2	8	0.25	20	14	1.43
1	2	0.5	2	6	0.33	10	19	0.53
1	2	0.5	2	4	0.50	185	336	0.55
1	2	0.5	4	6	0.67	0	6	0.00
1	2	0.5	4	4	1.00	39	90	0.43
2	1	2	2	8	0.25	20	14	1.43
2	1	2	2	6	0.33	6	70	0.09
2	1	2	2	4	0.50	49	234	0.21
2	1	2	4	6	0.67	0	62	0.00
2	1	2	4	4	1.00	36	195	0.18
2	2	1	2	8	0.25	20	14	1.43
2	2	1	2	6	0.33	13	19	0.68
2	2	1	2	4	0.50	109	213	0.51
2	2	1	4	6	0.67	0	62	0.00
2	2	1	4	4	1.00	60	130	0.46

C. Impact of feedback

Table IV gives the results of the simulation run with the above given algorithm for different loss bound ratios. The following values are assigned to the variables: total arrival rate, $\lambda=14$ and service rate, $\mu=20$ to maintain the load = 0.7; other values t_h , t_l , b_h and b_l are varied to obtain the different packet loss ratios. The ratio t_h / t_l represents the negative feedback action and shows the quantitative change in threshold for high priority packets with respect to low priority. The ratio b_h / b_l represents the level of packet loss at which the change in threshold should happen. Thus, these two ratios depict the ADPBS behavior in terms of the amount of feedback action and time when this action is to be taken. This table captures the number of consecutive packet lost (high and low priority both) and tabulates these values for different combinations of ratios.

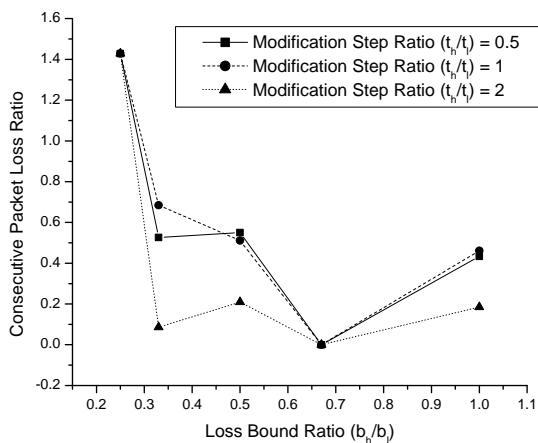


Fig. 9 Consecutive packet loss ratio for different loss bound ratios

Figure 9 gives the impact of feedback action using comparison among ratios of consecutive packets lost and loss bound ratios for various modification step ratios. It is observed that for lower loss bound ratios the packet loss ratio is higher while it decreases when the loss bound ratio is increased. For higher modification step ratios, the packet loss ratio further reduces.

V. CONCLUSION

This paper presents an Adaptive Partial Buffer Sharing (ADPBS) packet loss control scheme for two and multiple priority classes in congested networks. The proposed scheme incorporates an adaptive threshold which dynamically adjusts according to network traffic behavior changes. An expression relating packet loss ratio of two adjacent priority classes and four system control parameters is derived. The performance of ADPBS scheme is compared and analysed with SPBS and FIFO queues for different load ratios and input traffic combinations. ADPBS manages to reduce consecutive packet loss as compared to SPBS and FIFO queues due to its adaptive threshold nature. Our further research will focus on

enhancement of ADPBS scheme that delivers excellent performance under different traffic load conditions.

REFERENCES

- [1] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Fioroiu, "On achievable Service differentiation with token bucket marking for TCP", *Proceeding of ACM SIGMETRICS*, pp. 22-33, June 2000.
- [2] D.D Clark, W. Fang, "Explicit Allocation of Best Effort Packet delivery Scheme", *IEEE/ACM Transactions on Networking*, vol. 6, no.4, pp. 362-373, August, 1998.
- [3] J.W. Causey, H.S. Kim, "Comparison of buffer allocation schemes in ATM switches: Complete sharing, partial sharing, and dedicated allocation", *International Conference on Communications*, May 1994; Vol.2; pp.1164-1168.
- [4] D. Mitra and I.Ziedins, "Virtual partitioning by dynamic priorities: Fair and efficient resource-sharing by several services, *Broadband Communications: Network Services, Applications, Future Directions.*" *Lecture Notes in Computer Science*, Springer Verlag, 1996; pp. 173-185.
- [5] Cho, Jeong-Woo, Dong Ho, "Dynamic buffer management scheme based on rate estimation in packet-switched networks", *Computer Networks*, August, 2002; Vol.39, No.6; pp.769-787.
- [6] L.Georgiadis, I. Cidon, R. Guerin, and A. Khamisy, "Optimal Buffer Sharing", *IEEE J. Select. Areas Commun.*, Sept. 1995, vol. 13; pp. 1229-1240.
- [7] A. K. Choudhury and E.L. Hahne, "Space Priority Management in a shared Memory ATM Switch", in *Proc. IEEE GLOBECOM '93*, vol. 3, (Houston, Texas), Dec.1993; pp. 1375-1383.
- [8] C. Dovrilis, P. Ramanathan, "Proportional differentiated services, part II: Loss rate differentiation and packet dropping", *Proceeding of IWQoS*, June 2000; pp.52-61.
- [9] Czachowski T. and Peregrine F., "A queuing model for optimal control of partial buffer sharing", In *ATM Computer Operation Research*, Feb 98. 25(2): 113-126.
- [10] C. G. Kang and H.H. Tan, "Queuing analysis of explicit priority assignment partial buffers haring schemes for ATM networks", *IEEE/ACM INFOCOM*, March 1993; pp. 810-819.
- [11] G. Ascia, V. Catania, and D. Panno, "An Efficient Buffer Management Policy Based on An Integrated Fuzzy-Ga Approach", *IEEE/ACM INFOCOM*, June 2002.

Sakshi Kaushal received B.Tech. degree in Computer Science and Engineering from Punjab Technical University in 2001, and M.Tech. Degree in Computer Science and Engineering from Technical University, Haryana, India in 2002. and currently pursuing Ph.D at Thapar University, India.

She is currently Lecturer at Department of Computer Science and Engineering, Panjab University, Chandigarh, India. Her main research interest includes design, control, and performance analysis of broadband communication networks, queueing theory, and its applications to computer communications.

R. K Sharma received his PhD degree in mathematics from the University of Roorkee (Now, IIT Roorkee), India in 1993. He is currently professor in School of Mathematics and Computer application and Dean Academic Affairs at Thapar Institute of Engineering and Technology (TIET), Patiala India, where he teaches, among other things, queueing models and its usage in computer networks. His main research interests are in traffic analysis of computer networks, neural networks, and pattern recognition.