International Journal of Navigation and Port Research
Vol.27, No.3 pp. 283~288, 2003 (ISSN-1598-5725)

A Study on Performance Evaluation based on Packet Dropping in ATM Network : New Scheme Proposal

Seung-Seob Park[*] · *Dong-Cheol Yuk*[**]

[*] *Div. of Electronic, Computer and Telecommunication Engineering, PKNU, Busan Korea*
[**] *Graduate School of PKNU*

Abstract : *Recently, the growth of applications and services over high-speed Internet increases, ATM networks as wide area back-bone has been a major solution. As the conventional TCP/IP suite is still the standard protocol used to support upper application on current Internet, the issues regarding whether TCP/IP will operate efficiently on top of an ATM infrastructure and how to control its QoS still remain for studies.*

TCP uses a window-based protocol for flow control in the transport layer. When TCP uses the UBR service in ATM layer, the control method is only buffer management. If a cell is discarded in ATM layer, one whole packet of TCP will be lost; this fact occur the most TCP performance degradation. Several dropping strategies, such as Tail Drop, EPD, PPD, SPD, FBA, have been proposed to improve the TCP performance over ATM.

In this paper, to improve the TCP performance, we propose a packet dropping scheme that is based on comparison with EPD, SPD and FBA. Our proposed scheme is applied to schemes discussed in the previous technology. Our proposed scheme does not need to know each connection's mean packet size. When the buffer exceeds the given threshold, it is based on comparison between the number of dropped packet and the approved packet.

Our results are reported and discussed for comparing these discarding schemes under similar conditions. Although the number of virtual channel (VC) is increased, the simulation results showed that the proposed scheme can allocate more fairly each VC than other scheme.

Key words : *TCP/IP, UBR, ATM, EPD, SPD, FDA*

1. Introduction

As high speed networks proliferate, it is becoming apparent that no single technology will dominate all of the different networking environments. Indeed, it seems that ATM(Asynchronous Transfer Mode), Gigabit Ethernet, IP(Internet Protocol) switching will be co existed, interoperated. However, a common motivation for all high speed networking technologies continues to be the provision of QoS(Quality of Service), guarantees to the applications that they interconnect, though each technology has its own definition of QoS.

Further, a common issue is the application packet fragmentation and reassembly problem that severely affects application performance and network utilization.

While ATM was originally conceived as a carrier of integrated traffic, the recent momentum on the rapid standardization of the technology has come from data networking applications. Clearly, there is a need for ATM to support existing data applications. Since most data applications cannot predict their own bandwidth requirements, they usually require a service that dynamically shares the available bandwidth among all active users. Both ABR(Available Bit Rate) service and UBR(Unspecified Bit Rate) service have been considered for supporting data applications in ATM networks. The TCP(Transmission Control Protocol) is perhaps the most widely used transport layer protocol in existing data networks.

When TCP is run over ATM, TCP packets are segmented at the ATM layer into fixed-size small data units called cells. The ATM networks transfer user information cell by cell. When an ATM switch drops a cell because buffer overflows, the rest of the cell belong to the same packet of the discarded cell will still be transmitted. After the cells arrived at the destination, the destination fails to reassemble the packet to which the lost cell belonged. TCP has a retransmission mechanism to request the source for the corrupted packets. Therefore, once one or more cells constituting a packet are lost, the whole packet will be retransmitted; it means that the loss of one cell is amplified to the loss of a packet. And a portion of network resource is wasted to transmit the corrupted and useless packets. As the result, the loss and retransmission of TCP application data is responsible for most TCP performance degradation(Park Seung-Seob, 1999).

[*] Corresponding Author : Seung-Seob Park, parkss@pknu.ac.kr, 051) 620-6389
[**] net607@mail1.pknu.ac.kr

This phenomenon was observed by Romanow and Floyd. And two schemes named PPD(Partial Packet Discard) and EPD(Early Packet Discard) were proposed in Ref. (Cohen Reuven, 2000.) *to enhance the efficiency of TCP over ATM.* The performance of TCP over ATM using ABR and UBR services with EPD scheme is of major interest. Furthermore, different packet discard scheme were proposed to improve the fairness and the throughput of EPD algorithm.

Recently, some active queue management schemes are recommended to improve and preserve the performance of Internet in Ref. (Goyal R. 1997)(J Heinanen J, 1998).

The organization of the paper is as follows. In the next section, we describe the related work and discuss the underlying ideas of EPD, SPD and FBA. In Section 3, the operation of the proposed scheme will be described in details. Then simulation model and parameters are given in Section 4. In Section 5, the different existing schemes are quantitatively evaluated. Their result is shown. Finally, the concluding remarks are given in Section 6.

2. Various dropping policies

In this section, we describe a packet dropping schemes comparing with EPD, SPD and FBA.

2.1 TCP over UBR service

In its simplest form, an ATM switch implements a tail drop policy for the UBR service category as the earliest and easiest drop policy is called TD(Tail Drop) (Fang Chien 1994). This is also called TCP over plain ATM.

When a cell arrives at the FIFO queue, if the queue is full, the cell will be dropped, otherwise the cell will be accepted. If a cell is dropped, the TCP source loses the time waiting for the retransmission timeout, even though TCP congestion mechanisms effectively recover from loss, the resulting throughput can be very low. It is also known that simple FIFO buffering with tail drop of ATM cells results in the receipt of incomplete segments. When part of a segment is dropped at the switch, the incomplete segment is dropped at the destination during reassembly. In fact, in Ref. (Fang Chien 1994), it is shown that wasted bandwidth further reduces the TCP throughput and results in very poor performance (Cohen Reuven, 2000).

2.2 Early Packet Discard scheme

As described previously, Tail Drop does not provide a satisfactory performance for TCP traffic.

In EPD, the first cell of any incoming packet will be discarded when the total queue size at a switch exceeds the EPD threshold. Once the first cell of a packet is discarded, the remaining cells of the packet will also be discarded, even when the switch queue is reduced to below the EPD threshold. However, a cell will not be discarded by the algorithm, if the first cell of the same packet is not discarded, unless the entire buffer is full. Consequently, EPD algorithm may utilize more network bandwidth than a network with Tail Drop policy only, because EPD algorithm checks unnecessary transmission to the minimum. However, TCP suffers significant performance degradation in terms of fairness (Siu H. Li., 1994)(Li H., 1996).

2.3 Selective Packet Dropping scheme

A further improvement to packet dropping policy is SPD(Selective Packet Dropping), it is also called packet dropping based upon per-VC accounting. The basic idea of SPD is that it keeps track of the activity of each VC by counting the number of cells from each VC in the buffer.

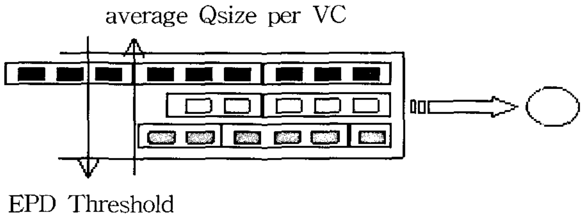

Fig. 1 Illustration of Selective Packet dropping

A fair allocation is calculated as the current buffer occupancy divided by number of active VCs.

Suppose VC_j has an incoming TCP packet to a switch. The first cell of the packet is discarded if $Q \geq Th$ for the current buffer size, and $Q_j \geq [Th]$ for VC_j, where Q_j is the number of cells from VC_j in the switch buffer, and $[Th]$ is defined as $[Th] = K \times Q/N$, N is the number of *active* VCs which have cells in the switch buffer, and K is a control parameter (typically, $1 \leq K \leq 2$). In fact, $[Th]$ represents the average buffer occupancy per VC when K=1.

Fig. 1 illustrates the SPD. Where we assume there are three VCs, the shaded of block area in the switch buffer indicates how much buffer a VC occupies. Note, however, that all VCs share a single FIFO queue. Thus, the emission of cells at the output will depend on the incoming order of cells to the buffer.

Fairness should be improved by using this per-VC accounting scheme, which aims at preventing each VC from occupying more than its fair share of the buffer. Moreover, at the switch output, the rate of cell emission should be fair among all contending VCs. However SPD is more complex and less scalable than the other policies. In addition to the EPD requirements, it also needs more state information to keep track of each VCs activity and more processing to calculate the fair share and weights.

2.4 Fair Buffer Allocation scheme

The other improvement to switch dropping policy is called FBA(Fair Buffer Allocation). It is also called per-VC queuing. This is similar to SPD and the same set of parameters as SPD. Now, a cell is dropped if the following conditions are met:

(C > threshold)
and
W(x) > Z × (B - Threshold)/(C - Threshold)

where B is the buffer size and C is the total amount of cells in the buffer. It uses a smooth form of parameter of Z and compares it with the load ration of VC.

In FBA, cells from different VCs will be put into different queues at each switch. Denote Q_j as the queue length for VC_j. When VC_j has a new packet coming, the first cell of the packet is discarded if $Q \geq Th$ for the total switch queue size (here Q denotes the sum of all VC buffer occupancies), and $Qj \geq [Th]$ for VC_j , where $[Th]$ is defined as $[Th] = K \times Q/N$, N is the number of active VCs which have cells in their queues, and K is a control parameter. All VC queues are served based on a round robin schedule.

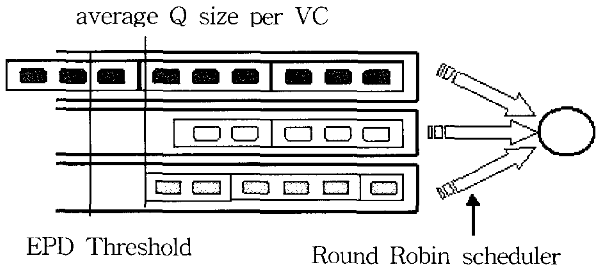

Fig. 2 Illustration of Fair Buffer Allocation

Fig. 2 shows the EPD with per-VC queuing mechanism. Each VC has a separate queue at the switch, while the shaded area indicates how large the queue is for each VC.

Our per-VC queuing scheme employs a round robin scheduler for the cell emission of the VCs, in addition to maintaining a fair buffer allocation. When the EPD threshold is reached, no VC exceeds its fair buffer share, and this guarantees fairness to some degree. Besides, in each round of cell emission, each VC can have only one cell output. Therefore, fairness can be achieved.

3. Our proposed scheme

When implementing a switch buffer, we can either make all VCs share a single FIFO buffer, or allow each VC to have its own FIFO buffer(Seu A., 2002). EPD, SPD, and FBA make use of single FIFO buffering. Therefore our scheme also implements to use a single buffer and two threshold, e.g., Max_threshold and Min_threshold.

The proposed scheme is illustrated by the flowchart in Fig. 3 and pseudocode in Fig. 4. We define the following parameters.

- Max_threshold = equal to EPD threshold
- Min_threshold = used to drop, if any cell is greater than min_threshold.
- Max = max value of the VciCount[VC]
- High selective control factor = value which used to determine any cell dropping or accepting with Max_threshold (Hscf)
- Low selective control factor = value which used to determine any cell dropping or accepting with Min_threshold (Lscf)
- VciCount[VC] = number of each VC's accepted packet

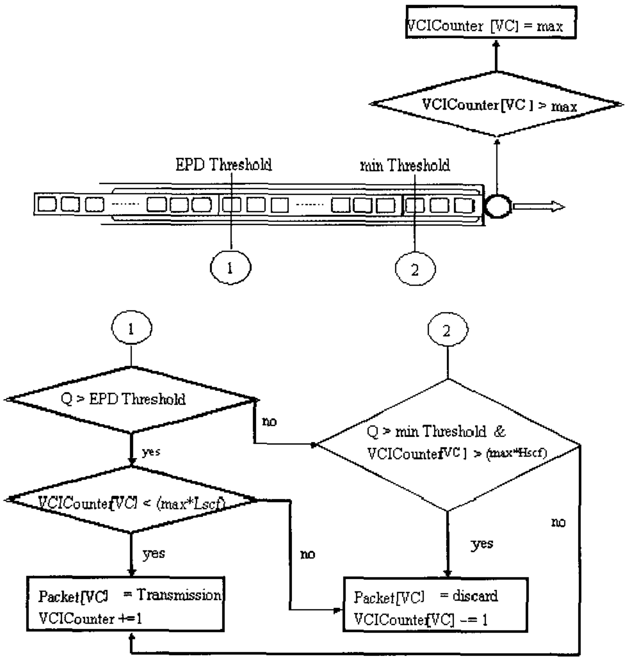

Fig. 3 Flowchart of proposed scheme

Note that Min_threshold determines that any cell of the packet is discarded, if the accepted count is greater than Max × High selective control factor using VciCount[VC]. VciCount[VC] is used to store the number of each VC's packet, if the discarded VC is decreased by one or the serviced VC is increased by one. The VciCount's element is set to be number of VC when each VC is initially connected. The Max′ value is updated to be the max value of VciCount.

SPD and FBA scheme have active VCs *N* and a control parameter *K*. Our scheme does not use active VCs information, but depends on VciCount of each VC.

```
When a cell is coming to an ATM switch:
  if the cell is the first cell of a packet
      if(Q ≥ max_threshold)
          if(VciCount[VC] < max/Low selective control factor)

                  VciCount[VC] ++;

                          accept the cell into the single FIFO queue;

        else
                        discard the cell;
                        VciCount[VC] --;
    else
    if(Q ≥ min_threshold)
          if(VciCount[VC] > max/High selective control factor)

                  discard the cell;

                  VciCount[VC] --;

                  else
                        VciCount[VC] ++;

                accept the cell into the single FIFO queue;
            else
              if( any cell of the packet has been discard )
                          discard the cell
                  else
                     if(Q ≥ Qmax)

                          discard the cell;
                     else
                          accept the cell into the single FIFO queue;
```

Fig. 4 Pseudocode of proposed scheme

A flowchart is shown in Fig. 3. As described previously in the EPD scheme, the first cell of a TCP packet is discarded if Q > Max_threshold, entire cells for new arriving frames are dropped to get over the network congestion, where Q is the number of cells in the switch buffer. But if VciCount[VC] is smaller than Max × Low selective control factor, the arriving cell is accepted in the relievable respect for fair buffer allocation. If VciCount[VC] is greater than Max × high selective control factor and Q ≥ Min_threshold, although any cell is currently in the switch buffer, that entire cells are discarded in the preventing monopolistic respect. At this time, VciCount[VC] is updated.

4. Simulation Model and Parameters

Our simulation tool is based on the YATS network simulator. YATS is an event-driven simulator composed of various components that send messages to each other (YATS 1997). We have built upon the package by adding SPD and FBA components.

Fig. 5 shows the network configuration used in our simulations consisting of two switches. All sources and destinations have *N* TCP connections and TCP flow control functions. We used a single FIFO queue with an output rate of 150 Mbps. This configuration has a single bottleneck link in the "backbone" shared by *N* UBR source.

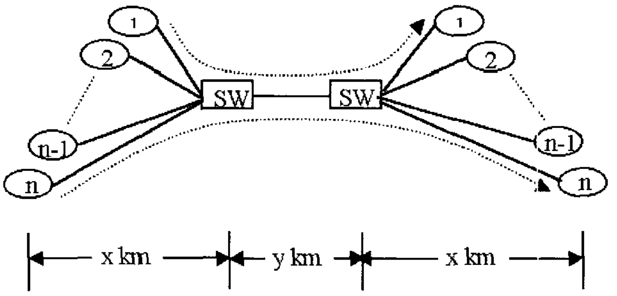

Fig. 5 Simulation network model

All traffic is unidirectional. A large infinite file transfer application runs on top of TCP for source. Ranges of *N* are from 5 to 70. We assume that the distance of between each TCP connection.

Table 1 Simulation parameters

- TCP Application
  - Data Frame = 512 bytes
    - Mean Packet Processing Delay = 200 usec
- TCP data
  - Mean Packet Processing = 300 us
  - Send/Receiver Buffer size = 64 Kbytes
  - Maximum segment size = 512 bytes
  - Default timeout = 500 msec
  - Minimum RTO = 100 msec
  - Maximum receiver window size = 64 Kbytes
  - Fast retransmission and recovery = on
- ATM UBR-Switch
  - Out Buffer size = 4096 cells
    - EPD Threshold = 3500 cells
- Link
  - Speed = 150 Mbps, Distance = 300 km

Our scheme is based on the improved EPD schemes using per-VC accounting and queuing techniques are based on the proposal in Heinanen and Kilkki (J Heinanen 1998). The parameters are employed in our simulations as follows Table 1.

The TCP packet processing delay is the time it takes the TCP source to handle the transmission of a data packet or the receipt of an acknowledgement packet. It also represents the time it takes the TCP destination to handle the receipt of a data packet or transmission of an acknowledgement packet.

Note that TCP default timeout will be used when the first packet of a connection is lost, since no round trip time can be applied to calculate the retransmission timeout.

5. Performance evaluation and Metrics

Through our simulations, the performance metrics for TCP over UBR are as follows:

- TCP effective throughput
- Fairness in terms of each scheme and proposed scheme

These performance metrics can be used to evaluate and compare the performance of different dropping policies.

As specified in Ref. (Jaussi L. 1998)(Goyal R., 1997), the performance of TCP can be measured by the efficiency, the fairness and the delay. We can define the TCP efficiency as: sum of TCP throughput/maximum possible TCP throughput. The maximum possible throughput is defined by the portion of the shared bottleneck link capacity that can be used by data traffic. For a given TCP Maximum size(MSS), the ATM Layer receives MSS + 40 bytes TCP/IP header + 8 bytes LLC/SNAP encapsulation + 8 bytes AAL5 header(total of 56). The AAL5 packet is segmented into $\lceil (MSS + 56)/48 \rceil$ cells. The maximum possible TCP throughput is given by (MSS × C) / ($\lceil (MSS + 56)/48 \rceil$ × 53(cells)) where C is the shared bottleneck link capacity.

One of the possible fairness metrics for TCP, is the fairness index introduced by Jain(see (Goyal R., 1997) for more details) and defined by $(\Sigma x_i)^2 / (n \times \Sigma x_i^2)$, where x_i is the throughput obtained by the connection 1 and n is the number of connections. Another fairness benchmark is the standard deviation which can be normalized by the expectation : $\sigma / \overline{x^1}$. When $\sigma / \overline{x}$ is close to 0, the dispersion of the distribution of x_i is around the mean, meaning that the algorithm is fair.

The simulation results are shown in Fig. 6. We performed experiments from 5 to 70 TCP connections using proposed scheme. Even with a large number of TCP connections, EPD does not significantly affect fairness in our simulation, and EPD shows large variance, because EPD does not perform selective discard of packets based on buffer usage.

And this is, as the number of source increase, the effect of a single source or a few sources on the fairness metric decrease. As a result, EPD might have a fairness value of 0.95 or better even if a few TCPs receive. On the contrary, proposed scheme provides higher fairness to a large number of sources than SPD and FBA's.

Ref. (Jain R., 1997) suggests that a value of 0.99 for the fairness metric reflects high fairness even for a large number of sources. Our scheme should provide high fairness in such cases, as shown in Fig. 7.

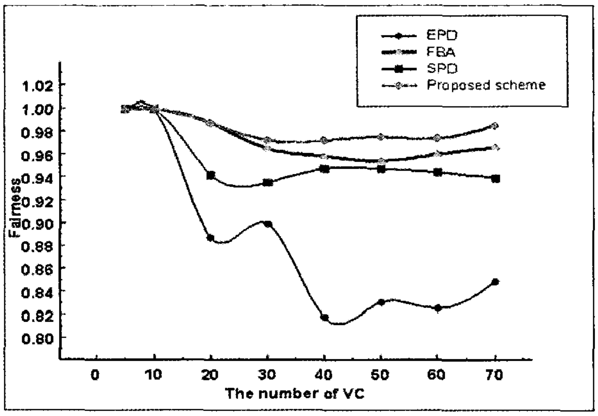

Fig. 6 Fairness in terms of the number of VC

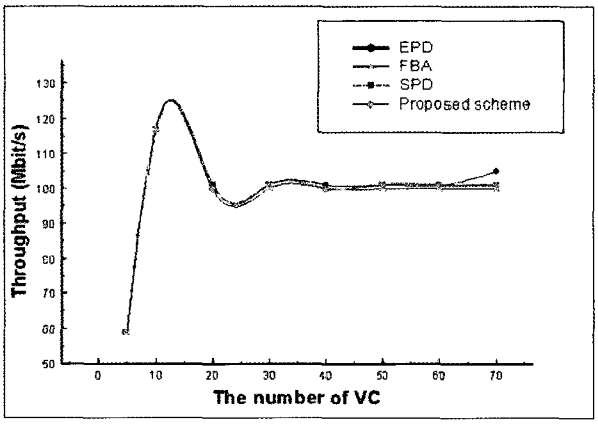

Fig. 7 Throughput in terms of the number of VC

For simulation parameters, we choose K = 1.0, Qmax = 4096 cells, and EPD Threshold = 0.8 * Qmax. We plotted TCP throughput. This results show that our scheme has a similar throughput such as SPD and FBA.

6. Conclusions

In this paper, we showed and analyzed simulation results of TCP traffic running over ATM networks with different dropping policies. The earliest proposal is to use a large buffer size without any dropping policies, but this is impractical. Tail drop just drops any incoming cell whenever the buffer is full, this results in the throughput degradation of TCP over ATM. EPD uses a threshold in the buffer to enhance for improving the throughput of TCP over ATM. The above polices have the unfairness problem.

Therefore, SPD and FBA have proposed for improving each connection's fairness. Our proposed scheme presented higher fairness than the other dropping schemes under similar conditions.

Acknowlegement

This work was supported by a grant of 2000 from PKNU. The authors wish to thank Dr. Youn at ICU in DaeJeon for fruitful discussion, which improved this paper.

References

[1] Cohen Reuven and Hamo Yaniv (2000), "Balanced Packet Discard For Improving TCP Performance in ATM Networks", pp.1566-1571, IEEE INFOCOM.

[2] Fang Chien, Chen Helen and Hutchins Jim (1994), "A simulation Study of TCP Performance in ATM Networks", pp. 1217-1223, GLOBELCOM Nov. Number 1.

[3] Goyal R., et al. (1997), "UBR+: Improving Performance of TCP over ATM-UBR Service," ICC '97, pp. 1042-1048, June.

[4] Goyal R., Jain R., Kalyanaraman S., Fahmy S., and Kim S. (1997), "UBR+ : Improving Performance of TCP over ATM- UBR Service", Proc. of IEEE ICC, pp.1042-1048.

[5] J Heinanen, K Kilkki (1998), "A fair buffer allocation scheme", Computer Communications, V.21, pp.220-226 Jan.

[6] Jain R. (1991), "The Art of Computer Systems Performance Analysis: Techniques For Experimental Design, Simulation, and Modeling," Wiley-Interscience. New York.

[7] Jaussi L., Lorang M., Nelissen J., (1998) "A Detailed Experimental Performance Evaluation on TCP over UBR," IEEE ICATM '98, pp.214-223, June.

[8] Li H., Siu K., Tzeng H., Ikeda C. and Suzuki H. (1996), "On TCP performance in ATM networks with per-VC early packet discard mechanisms" Computer Communication 19, pp. 1065-1076

[9] Park Seung-Seob, Yuk Dong-Cheol (1999), "Internet Protocols Over ABR and UBR Service : Problems, Approaches, and Their Evaluation," The trarsactions of KIPS, vol.6, no 11S, pp. 3260-3268(Korean), November.

[10] Romanow and Floyd S. (1995), "Dynamics of TCP traffic over ATM networks," IEEE, J. Selected Area in Comm, pp. 633-41, 13(4)(May).

[11] Seu A., Mohamoned (2002), "Fair Queuing with Round Robin : A New Packet Scheduling Algorithm for Routers", proc of IEEE ISCC'02, pp.1530-1536.

[12] Siu H. Li. and Tzeng H. (1994), "Intelligent congestion control for ABR service in ATM networks," Computer Comm. Rev, 24(5)(Octorber), pp.81-106

[13] (1997), YATS simulator for ATM networks, Dresden university of Technology.

Received 9 May 2003
Accepted 13 June 2003