Chapter 2

SD+: Improving TCP Performance over ATM-UBR Service Using Selective Drop Buffer Management Scheme

Aly E. Elabd and Mohamed A. Mostafa
Computer Engineering Department
Arab Academy for Science & Technology
Alexandria, Egypt

Key words: TCP over ATM, UBR, buffer management schemes, ATM traffic modelling

Abstract: A new packet discard scheme called Selective Drop with forgiveness mechanism that can be plugged into the existing Selective Drop buffer management scheme is developed. The new scheme that we call SD+ remarkably enhanced both throughput and fairness of TCP running over ATM-UBR service. Our scheme does not require per VC accounting but per VC-state where only one bit is needed. The focus here will be on demonstrating and analysing the effect of the forgiveness mechanism when plugged into the SD scheme in low latency networks. The SD scheme is selected because it is evolved as an improvement over both PPD and EPD policies that can be practically applied to existing ATM switches. This paper explains the concepts behind the new scheme and demonstrates its effect on TCP performance using simulation.

1. INTRODUCTION

ATM networks provide 6 different service categories: Constant Bit Rate (CBR), Variable Bit Rate (VBR-rt & VBR-nrt), Available Bit Rate (ABR), Guaranteed Frame rate (GFR) and Unspecified Bit Rate (UBR). UBR service is characterised by its simplicity, low overhead and cost effectiveness. It is also characterised by the lack of built in congestion

control mechanisms [5]. The ATM FORUM traffic management specification version 4.0 states that congestion control for UBR may be performed at a higher layer on an end-to-end basis. As 85% of computer networks are running TCP/IP [13], it is expected that many TCP implementations will use the UBR service category.

TCP employs a window based end-to-end congestion control mechanism to recover from segment loss and avoid congestion collapse thus providing a reliable and adaptive connection oriented service.

Several studies have analysed the performance of TCP over the ATM-UBR service concluding that TCP applications running over ATM-UBR switches with limited buffer size suffer from low throughput and are deprived from fairly sharing the available bandwidth [1, 7, 9,13]. Accordingly, several end system side and network side enhancements had been proposed to improve this situation.

Network side enhancements focused on improving the packet discard polices of ATM-UBR switches, aiming to avoid -as much as possible- hurting TCP throughput under network congestion conditions.

Network side drop policies can be roughly divided into two categories: simple discard policies and sophisticated buffer management policies. Simple discard policies like Partial Packet Discard (PPD) or Early Packet Discard (EPD) lead to enhanced throughput but low fairness. Sophisticated mechanisms like Selective Drop (SD), Random Early Detection (RED) and Fair Buffer Allocation (FBA) improve fairness and throughput [1,3] but require per VC accounting with considerable overhead and computations that increase algorithm complexity, in addition to being parameter sensitive.

In this paper, we propose the forgiveness mechanism that can be plugged into the existing buffer management schemes to remarkably enhance both throughput and fairness. Our mechanism does not require per VC accounting but per VC-state where only one bit is needed per VC. In a previous paper [15] we demonstrated the positive effect of our proposed forgiveness mechanism over EPD. Our focus her is demonstrating and analysing the effect of the forgiveness mechanism when plugged into the Selective Drop scheme in low latency networks to enhance further SD performance.

The paper is divided into eight sections; the first one is this introduction. Section two discusses the congestion control mechanisms in the TCP protocol. The third section describes the Selective Drop mechanism while section four presents our proposed forgiveness mechanism and describes the concepts behind it. Section five describes the simulation set-up used in all our experiments and defines the performance metrics. Section six illustrates the simulation results and section seven presents the paper conclusion. Finally, section eight is a summary and ideas for future work.

2. TCP CONGESTION CONTROL

TCP uses a window-based protocol for flow control that is enforced by two windows. The sender maintains a variable congestion window (CWND) as a measure of network capacity. The receiver maintains a receiver window (RCVWND) as a measure of its receiving buffer capacity. The number of bytes that the sender can send at a time is the minimum of the two windows.

The basic TCP congestion control scheme consists of the "slow start" and "congestion avoidance" phases as shown in figure (1). The variable SSTHRESH is maintained at sender side to distinguish between the two phases and is preliminarily set to 64 KB.

When a connection is established, the sender initialises the congestion window to the size of the maximum segment in use on the connection and sends one maximum segment. If the segment is acknowledged before a certain timer expires (retransmission timer), it adds a segment worth of bytes to the congestion window to make it two segments and then sends two segments. As each of these segments is acknowledged, the congestion window is increased by one maximum segment size. In effect, each burst successfully acknowledged doubles the congestion window. The congestion window keeps on growing exponentially until either the retransmission timer expires or the receiver's window is reached; this stage is called the slow start stage of TCP.

When a time out takes place (presumingly due to network congestion), SSTHRESH is set to half the size of the current CWND and CWND is set to one segment worth of bytes, a new slow start phase is starting now. When CWND is equal to SSTHRESH, the congestion avoidance phase is started in which CWND increases by one segment every round trip time; this results in a linear increase of CWND. Figure (1) illustrates TCP congestion control mechanism [1,11,13].

It is worth mentioning that the above description covers the standard TCP described in RFC 793. However, several enhancements have been proposed to avoid TCP congestion scheme described above. We choose the standard TCP implementation as it is still the most commonly used today.

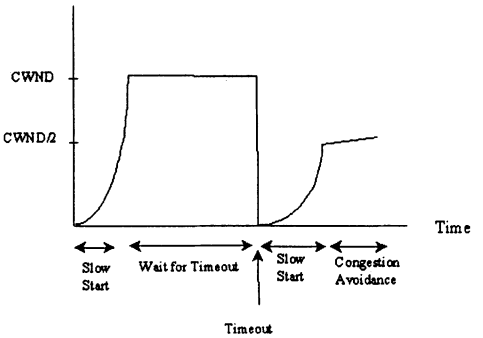

Figure 1. TCP congestion control

3. THE SELECTIVE DISCARD SCHEME

Selective Discard (SD) proposed by Goyal et Jain [1] is a buffer management scheme that can be classified among multiple accounting - single threshold (MA-ST) buffer management schemes [12]. SD evolved as an improvement over PPD and EPD policies.

PPD policy requires the switch to drop all subsequent cells from a packet if one cell has been dropped. EPD assigns a threshold smaller than buffer size, below which no cell discard takes place. If the queue length exceeds this threshold, EPD drops all arriving cells that belong to new packets while cells belonging to partially received packets are still accepted [9]. Both PPD and EPD do not guarantee fair sharing of available bandwidth between contending TCP sources [1,13].

SD ensures fair resources sharing by monitoring the share of each VC of the switch buffer. It deterministically drops cells belonging to greedy VCs in case buffer occupancy exceeds a pre-determined threshold R (similar to EPD threshold).SD keeps track of the activity of each VC by counting the number of cells from each VC in the buffer. A fair allocation is calculated as the (current buffer occupancy) divided by the (number of active VC's), where an active VC is defined as the VC with at least one cell in the buffer.

Let the buffer occupancy be denoted by X, and the number of active VCs be denoted by Na Then:

$$\text{Fair allocation} = X / Na \qquad (1)$$

The ratio of the number of cells of a VC in the buffer to the fair allocation gives a measure of how much the VC is overloading the buffer i.e., by what ratio it exceeds the fair allocation. Let Yi be the number of cells from VCi in the buffer, then the Load Ratio of VCi is defined as:

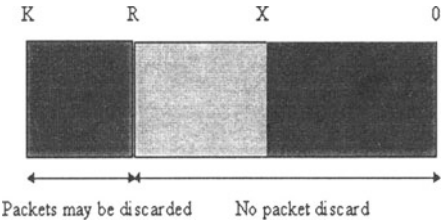

Figure 2. Selective Drop discard policy

Load Ratio of VCi = (Number of Cells from VCi) / (Fair allocation) = Yi*Na / X (2)

If buffer occupancy exceeds the static threshold R, and the load ratio of some VC is greater than a parameter Z, new packets from that VC are dropped, in preference to packets of another VC with load ratio less than Z. Thus Z is used as a cut off for the load ratio to indicate that a VC is overloading the switch. Figure (2) illustrates the buffer management policy of SD.

Goyal et Jain [1] carried an extensive set of simulations on SD concluding that the scheme is parameter sensitive (i.e. the achieved efficiency and fairness vary significantly with varying the values of Z and R). They proposed values of $Z= 0.8$ and $R= 0.9 * buffer\ capacity$ as the values that achieve the best results for efficiency and fairness.

4. SD+: THE FORGIVENESS MECHANISM

The forgiveness mechanism exploits a scheduling policy commonly used in operating systems. The scheduling algorithm tends to lower the priority of resources hungry processes in favour of lightly weighted processes. After a certain number of successive priority reductions, the scheduling algorithm forgets the past history of the heavy weighted process and treats it as a newly arriving job giving it the highest priority and a new chance to live [14].

We adapted the above concept to the ATM environment by assigning a single bit that we call the congestion bit to each VC. Whenever the congestion condition is met in SD switches (i.e. the buffer occupancy exceeds the static threshold R), newly arriving packets are continuously checked to see if they belong to greedy VCs (i.e. VCs that their buffer occupancy ratio exceeds the value of the parameter Z). If the arriving - packets belong to a greedy VC, they will be dropped and the congestion bit of its VC is set to one. In the next congestion episode, newly arriving packets belonging to VCs whose their congestion bit is set are not dropped, even if their share of buffer space exceeds the value of the parameter Z. This

time greedy VCs with clean congestion bit are dropped while greedy VCs with a set congestion bit are spared. The accepted VCs will have their congestion bit reset while the dropped VC's will have their congestion bit set. Thus forgiveness mechanism has no role during the first congestion episode but it takes over during next congestion. Hence, forgiveness mechanism accepts new cells from sources with lower value of CWND even when congestion condition is met.

Floyd et Jacobson [10] state that window flow control protocols have a periodic cycle equal to the connection round trip time, and that this periodicity can resonate (i.e. have a strong non-linear interaction) with deterministic control algorithms in network gateways. They also show that drop tail gateways in a TCP/IP network with strongly periodic traffic can result in systematic discrimination against some connections. Goyal et Jain [1] demonstrate the effect of such periodicity on TCP throughput and fairness in ATM networks with tail drop and early packet discard drop policies.

The forgiveness mechanism effectively helps reducing TCP synchronisation effects described in [1] and [10] and helps promoting randomness element in the network, hence breaks down phase effects and diminishes resonance responsible for efficiency degradation. This is accomplished by selectively and deterministically dropping connections with larger CWND while keeping other connections untouched in one congestion episode and reversing this action in the next one, the thing that will also enable TCP sources to exploit a fair share of the available network capacity in turn.

It is clear from section 2 that successive congestion badly hurt TCP efficiency. As the size of SSTHRESH will dramatically decrease leading to a situation in which the slow start phase, where CWND size increases exponentially, will be too short increasing the time taken by the TCP source to reach steady state after congestion breakdown. The forgiveness mechanism remarkably increases throughput of TCP sources as it prevents successive reduction of SSTHRESH.

The forgiveness mechanism requires minimal housekeeping overhead (one bit per VC). Also the logical condition it enforces can be easily applied in hardware which makes it compatible with ultra fast switching requirements of ATM networks and easily applicable in existing ATM switches.

As stated earlier the forgiveness mechanism can be plugged into several packet discard mechanisms to enhance the efficiency of TCP sources. We will show in this paper such effect in networks that apply SD switch buffer management scheme.

To evaluate the effect of the forgiveness mechanism, we use simulation, an approach commonly used in many other papers that dealt with the subject of traffic management of TCP/IP over ATM networks [1,2,3,7,9]. We use the ATM/HFC network simulator version 4.1, released December 1998, by the National Institute for Standards and Technology –NIST (USA). The simulator source code has been modified to add the forgiveness mechanism condition to the SD drop policy. Simulation parameters and performance metrics are described in details in the following section.

5. SIMULATION MODEL, PARAMETERS AND PERFORMANCE METRICS

5.1 Simulation model

All simulations presented in this paper use the N source configuration shown in figure (3). This model has been widely used in other papers discussing performance issues of TCP over ATM [1,2,3,7]. The model consists of N identical TCP sources that share a single bottleneck. The switches used in this model implement UBR service with either SD or SD+ (SD with the forgiveness mechanism plugged in).

5.2 Simulation parameters

Selection of parameters is in accordance with the guidelines in [8]. The following simulation parameters are used:

- The configuration consists of N identical TCP sources as shown in figure (3).
- All sources are ftp sources with very large file sizes (30 M B); ftp is a very common traffic pattern, large file sizes ensure continuos data flow throughout the simulation period.
- ATM applications are assumed to be running over SONET STS-3c links with bandwidth equal to 155.52 Mbps.
- The following link lengths have been tested: 0.8 Km, 1 Km, 2 Km and 3 Km per link.
- Switches are assumed to implement SD/SD+ discard policies; R parameter is set to *buffer size* 0.90* cells and Z parameter is set to 0.8;these values are selected because they ensure optimum results for efficiency and fairness as described in [1].
- Switch buffer sizes of 1000 and 3000 cells have been tested respectively.

- Peak cell rate is 146.9 Mbps (after factoring in SONET overhead).

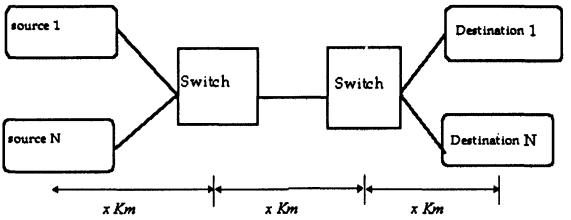

Figure 3. N-Source configuration

- Traffic is unidirectional, only sources can send data and destinations send acknowledgements only.
- TCP maximum segment size is set to 512 bytes; this is the standard TCP segment size. Testing with larger segment sizes is left to future work.
- TCP retransmission timer granularity is set to 100 ms.
- TCP RCVWND is set to 64 KB; this is the default TCP receiver window size.
- All TCP sources start sending data at the same time and keep on sending data throughout the simulation, this pattern increases the probability of TCP synchronisation.
- Simulated duration is 10 seconds.

5.3 Performance metrics

Two performance metrics are used: efficiency and fairness, the same set of metrics have been used in several studies for performance measurements of TCP over UBR [1,3,7]. They also conform to the recommendations of the test-working group of the ATM FORUM [6].

Efficiency can be defined as the ratio of the goodput achieved by all TCP

$$\sum_{i=1}^{i=N} \frac{Xi}{C} \qquad (3)$$

connections to the maximum achievable throughput of TCP over ATM-UBR, interpreting the above sentence to a mathematical formula, we get:
Where N is the total number of TCP sources, x_i is the goodput of TCP source number i and C is the maximum achievable throughput of TCP over a certain ATM link.

The goodput of a TCP source is defined as the total number of successfully transmitted bytes (excluding retransmissions) divided by the

simulation time. The maximum possible TCP throughput over UBR can be calculated as follows:

For a 512 bytes TCP segment size, ATM layer receives the following payload:

8 byte LLC header	20 byte IP header	20 byte TCP header	512 byte TCP payload	8 byte AAL5 trailer

Figure 4. Payload at the ATM layer

This payload is padded to form 12 ATM cells, thus 512 data bytes at the TCP layer results into 636 bytes at the ATM layer.

So, the maximum possible throughput is 512/636 =80.5 %=125.2 Mbps on a 155.52 Mbps link. For ATM over SONET, this number is further reduced to 120.5 Mbps [1,11]. Fairness is measured by the fairness index F defined by:

$$FairnessIn\,dex(F) = \frac{\left(\sum_{i=1}^{i=N} \frac{xi}{ei}\right)^2}{N * \sum_{i=1}^{i=N} \left(\frac{xi}{ei}\right)^2} \quad (4)$$

Where e_i is the fair share of source i of the available bandwidth for the N source configuration $e_i = C / N$. A fairness value of 0.9 may or may not be accepted according to the nature of application and the number of sources involved. A fairness index of 0.99 is considered to be near perfect [1].

6. SIMULATION RESULTS

We simulated 5 TCP sources with finite buffer switches. The simulations were performed using two values of buffer sizes 1000 and 3000 cells respectively, these values were chosen because they are closer to existing LAN switches and have been used in other studies [1]. The results are illustrated in the following tables:

Table 1. SD Vs. SD+ (Efficiency)

Buffer size (cells)	Link length(Km)	SD	SD+
1000	0.8	0.55	0.62
3000	***0.8***	***0.67***	***0.89***
1000	1	0.51	0.6
3000	***1***	***0.71***	***0.91***
1000	2	0.66	0.62
3000	***2***	***0.69***	***0.89***
1000	3	0.35	0.6
3000	***3***	***0.71***	***0.91***

Table 2. SD Vs. SD+ (Fairness)

Buffer size (cells)	Link length(Km)	SD	SD+
1000	0.8	0.98	1
3000	0.8	***0.97***	***0.99***
1000	1	0.87	0.99
3000	1	***0.99***	***0.99***
1000	2	0.99	0.98
3000	2	***0.97***	***0.99***
1000	3	0.85	0.94
3000	3	***0.97***	***0.99***

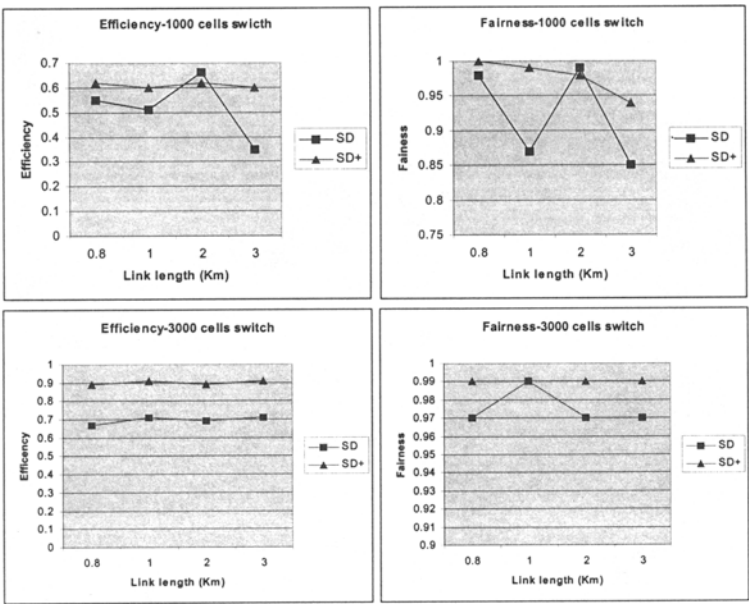

Figure 5. Performance of SD+ Vs. SD

Now, we want to examine the performance of SD+ in case of large number of sources; we simulated 10 TCP sources with finite buffer switches. The simulations were performed using two values of buffer sizes 2000 and 3000 cells respectively for link length of 0.8 Km, 1 Km and 1.5 Km; simulation results are illustrated below.

Table 3. SD+ Vs. SD for large number of sources (Efficiency)

Buffer size (cells)	Link length(Km)	SD	SD+
2000	0.8	0.73	0.93
3000	0.8	***0.32***	***0.93***
2000	1	0.54	0.94
3000	1	***0.93***	***0.94***
2000	1.5	0.60	0.89
3000	1.5	***0.96***	***0.93***

Table 4. SD+ Vs. SD for large number of sources (Fairness)

Buffer size (cells)	Link length(Km)	SD	SD+
2000	0.8	0.91	0.98
3000	0.8	***0.73***	***0.97***
2000	1	0.87	0.98
3000	1	***0.96***	***0.97***
2000	1.5	0.92	0.98
3000	1.5	***0.94***	***0.96***

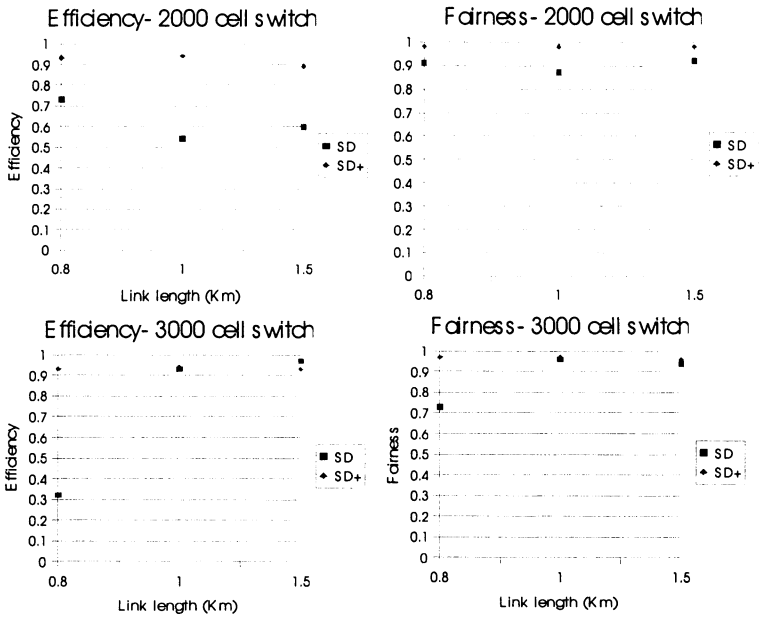

Figure 6. Performance of SD+ Vs. SD for large number of sources

From tables (1), (2), (3) and (4) the following observations can be made:

Buffer size is proportionally related to performance: It is clear that switches with larger buffer size lead to obvious increase in TCP efficiency and configuration fairness; under same network conditions. The switch with 3000 cells buffer capacity leads to 10% ~ 40% rise in the efficiency and 1% ~ 7% rise in fairness index than that of 2000 cells buffer size.

SD+ remarkably enhances performance of TCP over ATM-UBR: Remarkable increase in efficiency is observed when applying SD+. This improvement is due to SD+ drop policy that breaks synchronisation between TCP sources. Hence decreasing the probability of switch 's buffer overflow which negatively affect TCP throughput as it forces TCP sources to enter the slow start phase decreasing CWND to 1 as described in section 2. SD+ most remarkable efficiency improvement is achieved in the case of 2000 cell

switches network where about 48% increase in efficiency is observed while the highest efficiency value (0.94) is achieved in the network with 3000 cell buffer switches.

SD+ enhances fairness: SD+ increases efficiency but does not sacrifice fairness, fairness value achieved with SD+ is higher than that obtained when applying SD in most cases. Even when SD achieves a suitable fairness index (0.99) in networks with larger switch buffer sizes, it achieves the same value or even increases it to the ideal value (1). This is due to SD+ drop policy which fairly allocates the bandwidth between the contending sources by granting the available network resources to TCP sources with lower CWND.

SD+ has a steady state performance: It is clear from simulation results that SD+ has a steady state performance in all configurations simulated where the values of achieved efficiency and fairness are almost constant, while the performance of SD was fluctuating between good performance and unexpected performance breakdown.

7. CONCLUSION

SD+ enhances efficiency and fairness over SD. The positive effect is obvious in switches with lower buffer sizes (1000 & 2000 cell switches) where remarkable improvement is observed. In case of switches with 3000 cells buffer size, the achieved values of efficiency and fairness are almost equal to the values achieved by SD.

SD+ performance is stable in different network configurations while the performance of SD is fluctuating when tested under the same configurations.

8. SUMMARY & IDEAS FOR FUTURE WORK

TCP traffic phase effects and poor buffer management policies lead to TCP performance degradation in ATM-UBR networks. Simple drop policies like EPD considerably improve efficiency but it is not guaranteed to provide fairness. Sophisticated drop mechanisms like SD and FBA improve both efficiency and fairness much further using per VC accounting that require considerable accounting overhead and increased algorithm complexity, in addition to being parameters sensitive. Our proposed forgiveness mechanism can be used as a plug in to any of the above mentioned schemes to remarkably improve performance and fairness, we showed how the

forgiveness mechanism can improve the performance of SD yet requiring minimal overhead (1 bit per VC).

In this paper, we focused on analysing the effect of forgiveness mechanism over SD. In a future paper we will show how the forgiveness mechanism can considerably improve the performance of FBA, it is also interesting to investigate the effect of larger number of TCP sources and larger TCP segment sizes on TCP throughput when using SD+.

REFERENCES

[1] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy and Boby Vandalore, "Improving the performance of TCP over the ATM-UBR service," Journal of Computer Communications Vol. 21, No. 10, July 1998.

[2] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy and Rohit Goyal, "Performance and buffering requirements of Internet protocols over ATM ABR and UBR services", IEEE communications Magazine,June 1998

[3] Vincent Rosolen, Olivier Bonavnture and Guy Leduc, "Impact of cell discard strategies on TCP/IP in ATM UBR networks", Proc. of the 6th Workshop on Performance Modeling and Evaluation of ATM Networks (IFIP ATM'98) Ilkley, UK,July 98.

[4] Kenji Kawahara, Koichiro Kitajima, Ttetsuya Takine and Yuji Oie, "Packet loss Performance of Selective Cell Discard Schemes in ATM Switches", IEEE Journal on selected areas in communications, Vol. 15, NO. 5, July 1997.

[5] ATM FORUM, ATM Traffic Management Specification Version 4.0, Apr. 1996.

[6] Raj Jain and Gojko Babic, "Performance testing effort at the ATM FORUM : An Overview", IEEE Communications Magazine August, 1997.

[7] Allyn Ramanov, Sally Floyd, "Dynamics of TCP traffic over ATM Networks", IEEE Journal on Selected Areas In Communications, May , 1995.

[8] Tim Dwight, "Guidelines for the simulation of TCP/IP over ATM", ATM FORUM 95-0077r1, March 1995.

[9] Raj Jain, Rohit Goyal, Shiv Kalyanaraman, Sonia Fahmy and Fang Lu, "TCP/IP over UBR", ATM FORUM/96-0179.

[10] Sally Floyd and Van Jacobson, "On Traffic Phase Effects in Packet-Switched Gateways", Computer Communication Review,V.21N.2, April,1991.

[11] Andrew S. Tanenbaum. *Computer Networks: Third Edition, Prentic* -Hall, Inc., 1996.

[12] Rohit Goyal, "Traffic management for TCP/IP over Asynchronous Transfer Mode (ATM) Networks", Ph.D. Dissertation, Ohio-State University, 1999.

[13] Xiangrong CAI, "The performance of TCP Over ATM ABR and UBR Services", http"//www.cis.ohio-state.edu/~jain/cis788-97/tcp_over_atm/index.htm

[14] Harvey M. Deitel, *An Introduction to Operating systems: second edition*, Addision-Wesley publishing company, 1990.

[15] Aly E. El-Abd and Mohamed A. Mostafa, "EPD+: A New Packet Discard Mechanism to Improve TCP Performance over ATM-UBR", proceedings of ICATM2000.