TCP/IP TRAFFIC OVER ATM NETWORKS WITH ABR FLOW AND CONGESTION CONTROL

Liping An and Nirwan Ansari
Center for Comm. and Sig. Proc.
Dept. of ECE, NJIT
Newark, NJ 07102, USA

Ambalavanar Arulambalam
Bell Labs, Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ 07974, USA

ABSTRACT

In this paper, we compare the performance of TCP/IP traffic running on different rate based ABR flow control algorithms such as EFCI, ERICA and FMMRA by extensive simulations. The FMMRA algorithm is shown to exhibit the favorable features of least buffer requirement, fair bandwidth allocation to TCP connections, fast and accurate ACR rate adjustment according to the changes of network traffic, and the highest effective TCP throughput.

1. Introduction

The primary goal of ABR service is intended to economically support data applications which do not have explicit throughput and transmission delay requirement, such as file transfer (FTP) and remote login (TELNET). Most of the data applications cannot predict their own traffic parameters and have bursty nature. These applications can tolerate transmission delay, but very sensitive to data lost. The simulation study in [6] shows that the performance of TCP/IP over ATM networks without ATM level congestion control is quite poor when the switch is congested and begins dropping cells.

In order to achieve high bandwidth utilization and in mean while avoid the cell loss due to network congestion, some kind of ATM level flow and congestion control mechanism is necessary. After a considerable debate, a rate-based flow control framework for the ABR service had been specified by the ATM Forum [1]. In this framework, ATM switch is responsible for fairly allocating the bandwidth among all connections that compete at this switch point, and send this information back to the source end-system periodically using Resource Management (RM) cells. Since this allocation policy is implementation specific, it has been the focus of switch design and implementation for the last few years. This issue has been becoming one of the important differentiating factors for the next generation of commercially available ATM switches.

Many rate-based ABR flow and congestion control algorithms have been proposed, and extensive simulations have been done on the ABR level. A survey of nine proposed rate-based ABR algorithms is presented in [2]. However, few research literature has been found on performance comparison of TCP/IP traffic running on these algorithms. In this paper, we select three representative ABR algorithms: EFCI, ERICA and FMMRA to simulate the performance of TCP/IP running over them. EFCI is chosen because it is a simple binary scheme which is implemented in most of today's ATM switches. Its implementation cost is low, but it does not ensure fair rate allocation and exhibits the well known beat-down problem. ERICA is a typical Explicit Rate (ER) algorithm using congestion avoidance, and has a reasonable implementation complexity. FMMRA is a max-min rate based algorithm which has several advantages over other algorithms, but it has relatively higher implementation complexity.

2. TCP Traffic over ATM Networks

An important characteristic of a TCP congestion control algorithm is that it assumes no support from the underlying layers to indicate or control congestion, but instead it uses implicit signals such as acknowledgments, time-outs, and duplicate acknowledgments to infer the state of the network. These feedback signals are used to control the amount of traffic injected into the network by modifying the window-size used by the sender. The algorithm attempts to utilize the available bandwidth of the network as much as possible, without, at the same time, introducing congestion. The congestion control mechanisms used in TCP are based on a number of ideas proposed by Jacobson [3]. Most of today's TCP implementations are based on or derived from either 4.3 BSD UNIX Tahoe or Reno version.

The TCP congestion control mechanism consists of three parts: slow start, congestion avoidance, retransmission and exponential backoff. The slow-start algorithm is used to perform congestion recovery by decreasing the window-size to one segment, and doubling it once every round-trip time. The term slow-start may be a misnomer because the actual window size is increased exponentially. It takes only $log_2 N$ round trips to attain a window size of N segments. If there are multiple TCP connections connected to the same ATM switch, the traffic load could be increased very quickly. Slow start allows the TCP source to quickly attain maximum transmission rates when the network bandwidth is available. Once the congestion window reaches the slow start threshold, TCP enters the congestion avoidance phase, and slows down the rate of increment. The purpose is to probe for additional available bandwidth in the network and at the same time to avoid causing additional congestion. The retransmission and exponential backoff mechanism retransmits the packet after a packet loss is detected, and attempts to maintain a good estimate of the round-trip delay which is used as a basis to set the retransmission timers.

Each TCP packet is fragmented into many short 53-byte ATM cells. The longer the TCP packet, the more ATM cells are fragmented into. All these ATM cells are originated from TCP sources and multiplexed by the switches on the way to their destinations. Even if only one cell of the TCP packet is dropped by a congested switch, the whole packet becomes useless, and needs to be retransmitted. This is the well known TCP fragmentation problem over ATM network. Owing to this phenomenon, the ATM layer cell loss ratio due to congestion does not indicate the TCP throughput loss at all. One percent cell loss can cause 10% or even 50% ampli-

This work was supported in part by AT&T/Lucent Technologies Foundation through a special purpose grant.

fied throughput loss. Normally, the longer the TCP packet, the worse the performance of TCP due to congestion.

Most of today's TCP implementations use 0.5 second timer granularity. Compared to the Round Trip Time (RTT) of high speed, low delay ATM network, this timer granularity is too coarse. While TCP sources are waiting for the time out period, a considerable amount of time and bandwidth are wasted. A simulation result [7] showed that the TCP effective throughput over plain ATM network without any congestion control can be as low as 34% of the maximum possible.

TCP can achieve its maximum throughput only when there is no cell loss. The TCP packet length and window size, Round Trip Time (RTT) of the network, the switch buffer size, and the congestion algorithm are factors that contribute to the cell loss ratio. Although we can reduce some congestion by reducing the TCP packet length and window size, congestion caused by the high frequency burst background VBR traffic and long round trip delay cannot be eliminated. Also, simply reducing the TCP packet length and window size results in low transmission efficiency and link utilization. In order to achieve an acceptable TCP throughput performance, some kind of ATM layer congestion control algorithm implemented in ATM switches is necessary.

3. ABR Rate-based Flow Control Mechanism

In the ABR service, the source adapts its rate to network conditions. Information about the state of the network, such as bandwidth availability, state of congestion, and impending congestion, is conveyed to the source through special probe cells called Resource Management Cells (RM-cells). The scheme is based on a closed-loop, "positive feedback" rate control principle. Here, the source only increases its sending rate for a connection when given an explicit positive indication to do so, and in the absence of such a positive indication, continually decreases its sending rate.

The source generates RM cells in proportion to its current data cell rate. The destination will turn around and send back the RM cell to the source in the backward direction. RM cells which can be examined and modified by switches in both forward and backward directions carry the feedback information of the state of congestion and the fair rate allocation. A switch shall implement at least one of the following methods to control congestion at a queuing point: (1) Explicit Forward Congestion Indication (EFCI) marking in which the switch may set the EFCI state in the data cell headers, and most of the first generation switches had implemented this mechanism before the RM cell was fully defined; (2) Relative rate marking in which the switch may set the congestion indication (CI) bit or the no increase (NI) bit in forward and/or backward RM cells; (3) Explicit rate marking in which the switch may reduce the explicit rate (ER) field in forward and/or backward RM cells. Switches that implement options (1) and (2) are known as binary switches which can reduce implementation complexity but may result in unfairness, congestion oscillation, and slow congestion response. Switches that implement option (3) are generally called ER switches which require sophisticated mechanisms in place at switches for the computation of a fair share of the bandwidth. The standard-defined source and destination behaviors allow the inter-operation of the above three options. Details of the ATM Forum congestion control framework for ABR service are beyond the scope of this paper and can be found in [1].

Based on the ATM Forum's rate-based congestion control framework, many ABR algorithms with different performance and implementation complexity have been proposed in the past few years. Among these algorithms, the following three representative algorithms are studied for TCP over ATM in this paper: Explicit Forward Congestion Indication (EFCI), Explicit Rate Indication for Congestion Avoidance (ERICA), Fast Max-Min Rate Allocation (FMMRA) algorithm.

In an EFCI-based switch, if congestion is experienced in an intermediate switch during connection, the EFCI bit in the data cell will be set to 1 to indicate congestion. The CI field in the RM cell is set by the destination if the last received data cell has the EFCI field set and is returned back to the source. If the source receives an RM cell with no congestion indication, the source is allowed to increase its rate. If the congestion indication bit is set, the source should decrease its rate. The parameters RIF and RDF control the rate by which the source increases or decreases its rate. The EFCI-based switches suffer from a phenomenon called the beat-down problem. In a network using only EFCI-based switches, where a congested switch marks the EFCI bit of the data cell, sources traveling more hops have a higher probability of getting their cells marked than those traveling fewer hops. As a result, it is unlikely that these long-hop connections are able to increase their rates and consequently are beaten down by these short-hop connections.

The ERICA algorithm is an approximation fair rate computation and congestion avoidance algorithm. A switch is operated at a congestion avoidance status by specifying a less than 100% target link utilization factor. Normally this factor is chosen to be 0.9, implying that only 90% of the total bandwidth is available to ATM connections and the remaining 10% is used to drain the ABR queue when sustained congestion occurs. Instead of directly calculating the max-min fair rate, the switch calculates the fair-share rate for each VC connection. If any VC connection cannot use the fair-rate due to bottlenecked elsewhere, in the next round trip time, the switch will experience a traffic load below the target link utilization. When under utilization is detected at a switch, the unused bandwidth is reallocated to the unbottlenecked VC connections, and the traffic load will hopefully, after several round trip times, converge to the target link utilization, and each individual VC connection will reach its max-min fair rate. Since ERICA algorithm operates in a congestion-avoidance state, it is insensitive to parameter variations, and proves to be very robust. Also because it does not have to keep bottleneck information for each VC connection, the switch implementation is relatively simple compared to other ER algorithms. This algorithm, however, has some limitations in achieving desired fairness for all the connections and buffer requirements. In some cases, a connection that gets started late, though acquiring its equal link share, may not get the max-min rate. For complete details of the algorithm, the reader is referred to [4], [5].

The Fast Max-Min Rate Allocation (FMMRA) [2] algorithm is based on measurement of available capacity and exact calculation of max-min fair rates. Each ABR queue in the switch computes a rate that it can support. This rate is

0-7803-4201-1/97/$10.00 (c) 1997 IEEE

referred to as the advertised rate. The advertised rate along with the ER field in the RM cell are used to determine if the connection is bottlenecked elsewhere. If a connection cannot use the advertised rate, it is marked as a bottlenecked elsewhere and its bottleneck bandwidth is recorded. The ER field in the RM cell is read and marked in both directions to speed up the rate allocation process. The bi-directional ER marking in this algorithm makes it possible for downstream switches to learn bottleneck bandwidth information of upstream switches, and the upstream switches to learn bottleneck bandwidth information of the downstream switches. Many of the proposed algorithms mark the ER field only in the backward direction. Because of the uni-directional ER marking, switches closer to the source get more accurate ER information than those closer to the destination. This may result in slower response to congestion. This bi-directional updating of ER in the RM cell plays a significant role in drastically reducing the convergence time of max-min fair rate allocation process. Details of FMMRA can be found in [2].

4. Simulations and Observations

Fig. 1 shows the network configuration used in our simulations consisting of two switches, N TCP sources and destinations, and one background VBR traffic. All links run at 155 Mbps. This configuration has a single bottleneck link in the "Backbone" shared by N ABR sources and one VBR source. A large infinite file transfer application runs on top of TCP for sources. Configurations with the "Backbone" link length of 1km represent typical Local Area Network (LAN) situations. The VBR background traffic is running at 100Mbps rate which is 2/3 of the total bandwidth. It starts at t=300 ms and is an ON/OFF burst source. Different ON/OFF frequencies are simulated. At the switch, VBR is given higher priority than ABR. If a VBR cell arrives at the switch, it will be scheduled for output before any awaiting ABR cells are scheduled. Because of limited link bandwidth, when the VBR is activated at t=300 ms, the switch will experience a congestion. In order to avoid buffer overflow, it then sends RM cells back to the source, and informs the ABR source to reduce its transmission rate. Different ABR congestion control algorithms will result in different buffer requirements and TCP performance.

We used an infinite source mode at the application layer running on top of TCP, implying that TCP always had a packet to send as long as its window permitted it. The purpose is to explore the possible limitations that ATM networks placed on the performance of the TCP protocol. The source TCP and ATM layer SES parameters were chosen as follows:

Source TCP Parameters
TCP maximum segment size = 9180 bytes
Mean packet processing time = 200 μs
Packet processing time variation = 50 μs
Receive window size = 64 K bytes
Bit rate = 155 Mbit/s
Delay-ack timer = 0

SES parameters
Peak cell rate = 155 Mbits
N_{rm} = 32 cells
M_{rm} = 2 cells
ICR = 10 Mbits
MCR = 0
CRM = TBE/N_{rm} = 20 cells
CDF = 0.5
TRM = 100 ms
TCR = 0.00424 Mbits

For ERICA and FMMRA, RDF = 1/512, RIF =1; for the binary scheme EFCI algorithm, RDF = 1/16, RIF = 0.1. For ERICA, the Target Utilization Factor was set at 90%, a level recommended by the proposer, but the Target Utilization Factor for FMMRA was set at 100% since FMMRA aims to achieve 100% utilization.

In our simulations, the following TCP and ABR performance metrics were evaluated:

- ABR queue length in the congested switch;
- TCP effective throughput;
- Link utilization at the congestion point;
- Source Allowed Cell Rate (ACR);

From the simulation results, the following observations were obtained:

1.Among the three ABR congestion control algorithms, FMMRA has the minimum buffer requirement for zero cell loss ratio. Fig. 2 shows the simulation results of ABR queue length vs. time for all of the three algorithms where two TCP sources and one VBR background traffic were employed. We find that, under the same network configurations, ATM switches implemented with FMMRA has the smallest ABR queue length, implying that FMMRA has the least buffer requirement for zero cell lost.

2. FMMRA has the best performance in fairly allocating available network bandwidth to individual TCP sources. Fig. 3 shows the results of the effective TCP throughput vs. time for the three algorithms. There were five TCP sources and one VBR background traffic with 10 ms ON and 10 ms OFF bursty nature. Both ERICA and EFCI lead to some unfairness among the individual TCP sources during the transient period. Some sources acquire higher throughputs than the others. Using FMMRA, every TCP source achieves the same effective throughput.

3. FMMRA has the fastest and most accurate response to the source ACR rate in response to the changes in network traffic. Fig. 4 shows results of ABR source Allowed Cell Rate (ACR) vs. time for the three algorithms. There were two TCP sources and one VBR background traffic with 100 ms ON and 100 ms OFF bursty nature. FMMRA has the fastest and most accurate response to the changes of available network ABR capacity.

4. Fig. 5 shows the effective TCP throughput of the three algorithms under the limited buffer size condition. With a switch buffer size of 1500 cells, FMMRA can achieve zero cell loss, hence yielding the highest throughput. We find there is a severe unfairness of bandwidth allocation among TCP connections for the EFCI algorithm. This is because EFCI only provides source with binary feedback

0-7803-4201-1/97/$10.00 (c) 1997 IEEE

instead of calculated fair rate. Once a TCP source experiences a cell loss and enters the slow start stage, other TCP sources take this advantage and increase their rates. When the next congestion occurs, this TCP source compete with others at an unfavorable situation. We also find there is a big throughput loss for the ERICA algorithm because of the cell loss in the switch.

5. Conclusions

In this paper we have presented and analyzed simulation results of TCP/IP traffic running over ATM network with different ABR congestion control schemes. From simulation results and the analysis, among EFCI, ERICA and FMMRA, under severe network congestion conditions, FMMRA, our recently proposed algorithm, exhibits the following favorable features compared to the other two algorithms:

- The least buffer requirement for zero cell lost;
- Fairly allocate available network bandwidth to individual TCP connections;
- Adjust source ACR rates fast and accurately in response to the changes of network traffic.
- Under limited switch buffer size situation, FMMRA achieves the best TCP throughput.

REFERENCES

[1] "The ATM Forum Traffic Management Specification," *The ATM Forum Specification*, April 1996.

[2] A. Arulambalam, X. Chen, and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks," *IEEE Communications Magazine*, vol. 34, no. 11, pp. 92–100, Nov. 1996.

[3] V. Jacobson, R. Braden, and D. Borman, "Congestion Avoidance and Control,," *Proceedings of Sigcomm'88*, pp. 314–329, 1988.

[4] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanthan, "ERICA Switch Algorithm: A Complete Description," *The ATM Forum Contribution 96-1172*, Aug. 1996.

[5] R. Jain, S. Kalyanaraman, and R. Viswanthan, "Explicit Rate Indication for Congestion Avoidance," *The ATM Forum Contribution*, Nov. 1995.

[6] S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy, F. Lu, and S. Srinidhi, "Performance of TCP/IP over ABR," *Proceedings of Globecom'96*, London, vol. 1, pp. 468–475, Nov. 1996.

[7] A. Romanow and S. Floyd, "Dynamics of TCP Traffic over ATM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 4, pp. 633–641, May 1995.

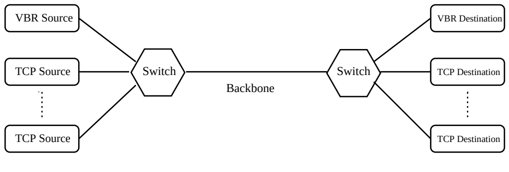
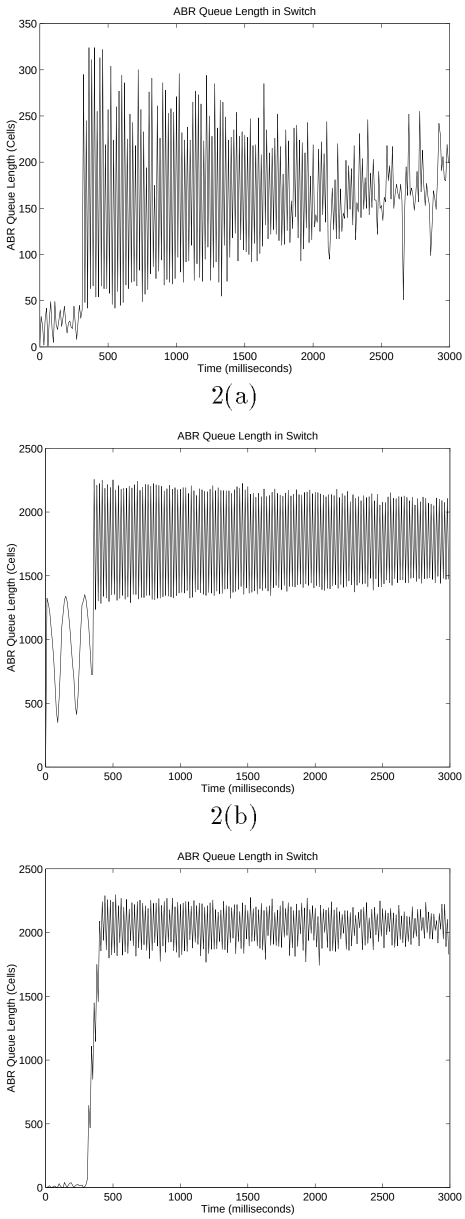


Fig. 1. Network configuration

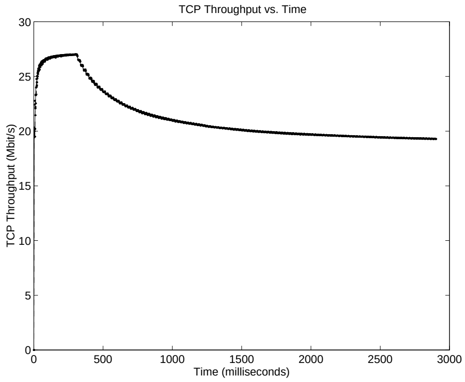


2(a)

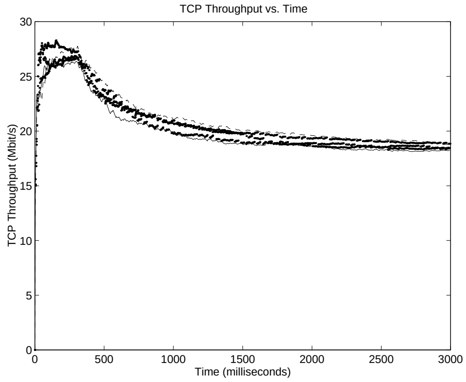
2(b)

2(c)

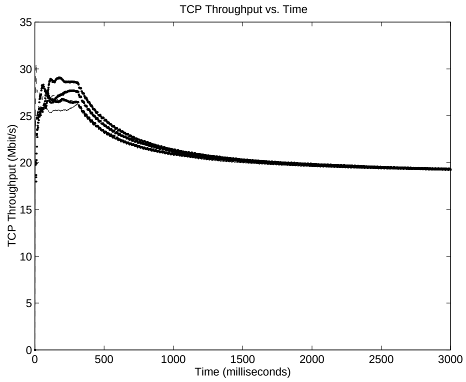
Fig. 2. ABR queue length using (a) FMMRA, (b) EFCI, and (c) ERICA, with 2 TCP & 1 VBR connections



3(a)

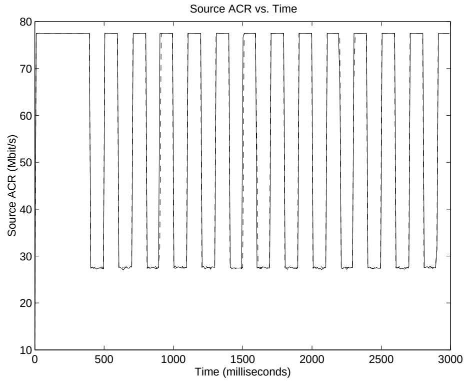


3(b)

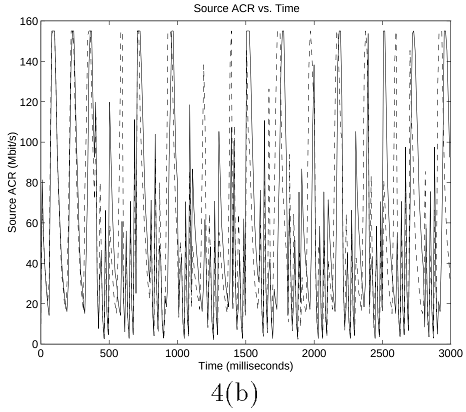


3(c)

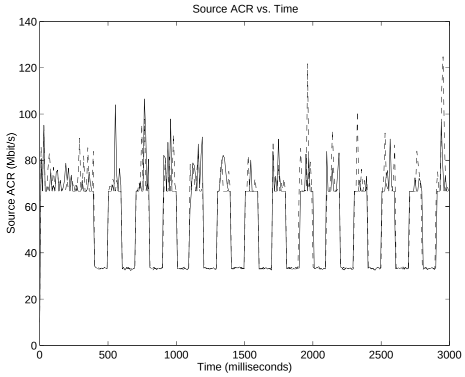
Fig. 3. TCP effective throughput using (a) FMMRA, (b) EFCI, and (c) ERICA, with 5 TCP & 1 VBR connections



4(a)

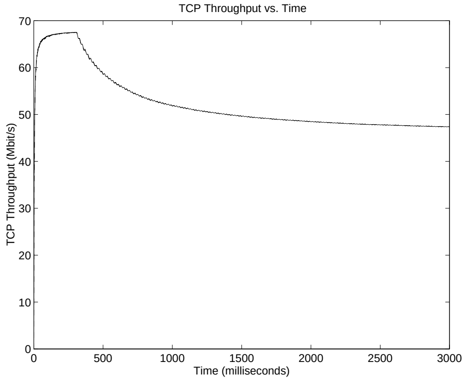


4(b)

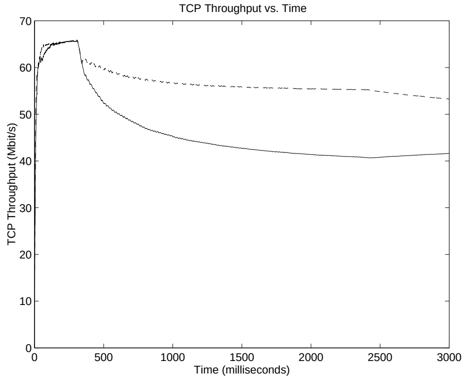


4(c)

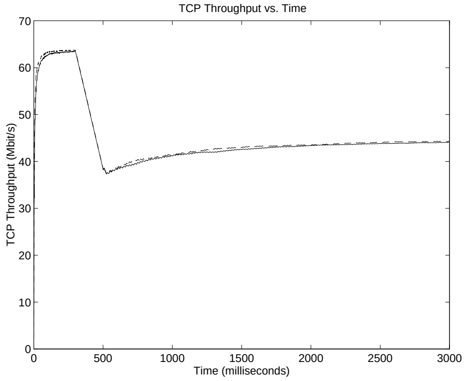
Fig. 4. TCP source Allowed Cell Rate(ACR) using (a) FMMRA, (b) EFCI, and (c) ERICA, with 2 TCP & 1 VBR connections



5(a)



5(b)



5(c)

Fig. 5. Effective TCP throughput using (a) FMMRA, (b) EFCI, and (c) ERICA, with imited buffer size of 1500 cells, 2 TCP & 1 VBR connections