Buffer Management and Rate Guarantees for TCP/IP over Satellite-ATM Networks[*†]

Rohit Goyal, Raj Jain, Sonia Fahmy, Bobby Vandalore, Mukul Goyal
Department of Computer and Information Science, The Ohio State University
395 Dreese Lab, 2015 Neil Avenue
Columbus, OH 43210-1277, U.S.A.
E-mail: goyal@cis.ohio-state.edu

Sastri Kota
Lockheed Martin Telecommunications
1272 Borregas Ave, Sunnyvale, CA 94089

Thomas vonDeak
NASA Lewis Research Center, MS: 54-6
21000 Brookpark Road
Cleveland, OH 44135

Abstract: *In this paper we present tehniques to to improve the performance TCP/IP over satellite-ATM networks, and to provide minimum rate guarantees to VCs carrying TCP/IP traffic. Many future systems are proposing to use ATM or ATM like technology to transport TCP/IP based data. These systems must be designed to support best effort services, as well as enhanced services that provide minimum rate guarantees to TCP/IP traffic. In this paper we study four main issues. First, we discuss tehniques for optimizing TCP performance over satellite networks using the Unspecified Bit Rate (UBR) service. We then demonstrate that high priority background traffic can degrade TCP performance over UBR, and we discuss the use of rate guarantees to improve performance. We design a full factorial experiment to assess the buffer requirements for TCP over satellite-ATM networks. Finally, we discuss the use of the Guaranteed Frame Rate (GFR) service category to provide minimum rate guarantees to VCs carrying TCP/IP traffic. We propose the Differential Fair Buffer Allocation (DFBA) scheme for buffer management that provides GFR guarantees to TCP/IP traffic over satellite latencies. We use full factorial experimental design with various latencies, buffer sizes, and TCP types for our simulations.*

Keywords: TCP, ATM, UBR, GFR, Buffer Management, Satellite

[*]This work was partially sponsored by the Nasa Lewis Research Center under contract number NAS3-97198

[†]Some results in section 3 have appeared in [6]. Results in section 5 have appeared in [7]. This paper is a much enhanced and consolidated version, and no results have been published in a journal.

1 Introduction

The TCP over Satellite (TCPSAT) working group in the IETF has designed mechanisms for transporting TCP over satellite networks. The group has focussed its efforts on modifying the TCP protocol so that its performance over satellite links is impro the various TCP and ATM enhancements, and discuss performance results for these experiments. Based on the experimental results and analysis, we provide guidelines for designing satellite-ATM network architectures that can efficiently transport TCP/IP data.

The paper does not propose any new TCP enhancements, but analyzes the performance of existing and proposed TCP mechanisms including Vanilla TCP (with slow start and congestion avoidance) TCP Reno (with fast retransmit and recovery), and TCP SACK (with selective acknowledgements). A performance analysis of TCP New Reno is presented in [14]. In this paper, we present techniques for buffer management, and throughput guarantees using ATM that improve TCP throughput, and are used to provide rate guarantees over satellite-ATM networks. The simulation and analysis are performed for various satellite latencies covering multihop LEO, and GEO systems. The results show that the design considerations for satellite networks are different than those for terrestrial networks, not only with respect to TCP, but also for the network. Several recent papers have analyzed various TCP policies over satellite latencies. These have been listed in [18]. The emphasis on network design issues for traffic management and basic service guarantees for TCP/IP over satellite-ATM is the unique contribution of this research.

2 Problem Statement: TCP over Satellite-ATM

There are three ATM service categories that are primarily designed for best effort data traffic. These are:

Unspecified Bit Rate (UBR): UBR is a best effort service category that provides no guarantees to the user. Past results have shown that TCP performs poorly over UBR. Two main reasons for the poor performance are the coarse grained TCP transmission timeout and TCP synchronization [15]. The performance of TCP over UBR can be enhanced by intelligent drop policies in the network. These drop policies include Early Packet Discard (EPD), Random Early Discard (RED), Selective Drop (SD) and Fair Buffer Allocation (FBA). These are discussed in [16]. Providing minimum rate guarantees to the UBR service category has also been suggested as a means for improving TCP performance over UBR. In this paper, we will analyze the UBR enhancements for TCP over satellite. The enhanced version of UBR has been informally termed UBR+.

Guaranteed Frame Rate (GFR): GFR is a frame based service that provides a Minimum Cell Rate (MCR) guarantee to VCs. In addition to MCR, GFR also provides a fair share of any unused network capacity. Several implementation options exist for GFR, including, network policing, per-VC scheduling, and intelligent buffer management. In this paper we show how to implement the GFR service using a buffer management algorithm called Differential Fair Buffer Allocation (DFBA). We discuss the performance of DFBA for TCP over satellite-ATM networks.

Available Bit Rate (ABR): The ABR service provides an MCR guarantee to the VCs, and a fair share of any unused capacity. ABR is different from GFR in several ways, but the most important is that ABR uses a rate based closed loop feedback control mechanism for congestion control. ABR also the feedback control to be end-to-end, or be broken into several hops using the virtual source/virtual destination option (VS/VD). A complete description and analysis of ABR and VS/VD is presented in [17]. In in this paper, we focus on TCP performance over UBR and GFR services.

The design of a multiservice satellite network present several architectural options for the ATM network component. These include the choice of the various ATM service categories and their implementations. We study the following design options for a satellite-ATM network supporting efficient services to transport TCP data:

UBR with tail drop or frame based discard like EPD. Among frame based discard policies, the Early Packet Discard [10] policy is widely used [24]. The Early Packet Discard maintains a threshold R, in the switch buffer. When the buffer occupancy exceeds R, then all new incoming packets are dropped. Partially received packets are accepted if possible. It has been shown [16] that for terrestrial networks, EPD improves the efficiency of TCP over UBR but does not improve fairness. We will examine the effect on EPD on satellite latencies.

UBR with intelligent buffer management. The Selective Drop scheme is an example of an intelligent buffer management scheme. This scheme uses per-VC accounting to maintain the current buffer utilization of each UBR VC. A fair allocation is calculated for each VC, and if the VC's buffer occupancy exceeds its fair allocation, its subsequent incoming packet is dropped. The scheme maintains a threshold R, as a fraction of the buffer capacity K. When the total buffer occupancy exceeds R×K, new packets are dropped depending on the VC_i's buffer occupancy (Y_i). In the Selective Drop scheme, a VC's *entire packet* is dropped if

$$(X > R) \text{ AND } (Y_i \times N_a / X > Z)$$

where N_a is the number of active VCs (VCs with at least one cell the buffer), and Z is another threshold parameter ($0 < Z \leq 1$) used to scale the effective drop threshold. In terrestrial networks, SD has been shown to improve the fairness of TCP connections running over UBR [16]. However, the effect of SD over satellite network has not been studied.

UBR with guaranteed rate allocation. A multiservice satellite network will transport higher priority variable bit rate traffic along with UBR traffic. The effect of higher priority traffic on TCP over UBR has not been studied before. Our simulations will show that higher priority traffic can degrade TCP performance in some cases. The results will also show, how rate guarantees to UBR can improve TCP performance in the presence of higher priority traffic.

GFR with buffer management, policing or scheduling. The GFR service enables the support of minimum rate guarantees to data traffic, and can be used to provide basic minimum rate services to data traffic. Currently very few suggested implementations of the GFR service exist. Sample implementations can use a combination of policing, buffer management and scheduling in the network. We will describe a buffer management scheme callled Differential Buffer Management (DFBA) scheme that can be used to implement the GFR service.

In addition to the network based options, there are four TCP congestion control techniques that are of interest in performance analysis over satellite links [18]:

Slow start and congestion avoidance (TCP Vanilla)

Fast retransmit and recovery (TCP Reno)

TCP New Reno

Selective acknowledgements (TCP SACK)

Vanilla and Reno TCP are standard mechanisms that are widely deployed in TCP stacks. TCP New Reno and SACK have recently been proposed as performance enhancements to TCP congestion control, and are being incorporated in TCP implementations. Several studies have reported performance results of the above TCP options over satellite latencies [18]. However, these studies have focussed only on TCP mechanisms, and have not considered intelligent network based traffic management and guaranteed rate policies. Also, the studies are all performed using a best effort service framework. Future broadband satellite networks must support the multiservice IP framework being adopted for terrestrial networks. Satellite networks usn an ATM based cell transport must use network based techniques to provide the service guarantees required for a multiservice network.

In this paper, we address the following components of optimizing TCP performance over Satellite-ATM networks:

Part 1 (Optimizing the performance of TCP over satellite-UBR) *We study the performance of TCP vanilla, TCP Reno, and TCP SACK, with buffer management policies within a best effort framework.*

Part 2 (Effect of higher priority traffic on TCP.) *We show how the performance of TCP degrades in the presence of higher priority traffic sharing the link. We also describe the use of guaranteed rate to improve TCP/UBR performance in the presence of higher priority traffic.*

Part 3 (Buffer requirements for TCP over satellite-UBR.) *We present simulation results to calculate the optimal buffer sizes for a large number of TCP sources over satellites.*

Part 4 (Performance of FR over satellite.) *We describe the GFR service category, and propose the DFBA scheme that uses a FIFO buffer and provides per-VC minimum rate guarantees to*

TCP traffic.

2.1 Performance Metrics

When ATM networks carry TCP/IP data, the end-to-end performance is measured at the TCP layer in the form of TCP throughput. To measure network performance, the throughputs of all TCPs passing through the bottleneck link are added, and expressed as a fraction of the total capacity of the bottleneck link. This is called the *efficiency* of the network. We now define this formally.

Let N TCP source-destination pairs send data over a network with bottleneck link capacity R bits/sec. Let x_i be the observed throughput of the ith TCP source ($0 < i < N$). Let C be the maximum TCP throughput achievable on the link. Let E be the efficiency of the network.

Definition 1 (Efficiency, E) *The Efficiency of the network is the ratio of the sum of the actual TCP throughputs to the maximum possible throughput achievable at the TCP layer.*

$$E(x_1, \ldots, x_N, C) = \frac{\sum_{i=1}^{i=N} x_i}{C}$$

The TCP throughputs x_i's are measured at the destination TCP layers. Throughput is defined as the total number of bytes delivered to the destination application (excluding retransmission and losses) divided by the total connection time. This definition is consistent with the definition of *goodput* in [23]

The maximum possible TCP throughput C is the throughput attainable by the TCP layer running over an ATM network with link capacity R. For example consider TCP over UBR on a 155.52 Mbps link (149.7 Mbps after SONET overhead). with a 9180 byte byte TCP MSS. For 9180 bytes of data, the ATM layer receives 9180 bytes of data + 20 bytes of TCP header + 20 bytes of IP header + 8 bytes of LLC header + 8 bytes of AAL5 trailer. These are padded to produce 193 ATM cells. Thus, each TCP segment results in 10229 bytes at the ATM Layer. From this, the maximum possible throughput = 9180/10229 = 89.7% = 135 Mbps approximately. It should be noted that

ATM layer throughput does not necessarily correspond to TCP level throughput, because some bandwidth may be wasted during TCP retransmissions.

In addition to providing high overall throughput, the network must also allocate throughput fairly among competing connections. The definition of fairness is determined by the particular service guarantees. For example, although UBR makes not service guarantees, fairness for TCP over UBR can be defined as the ability for UBR to provide equal throughput to all greedy TCP connections. In ABR and GFR, fairness is determined ability to meet the MCR guarantee, and to share the excess capacity in some reasonable fashion. We measure fairness using the Fairness Index F.

Definition 2 (Fairness Index, F) *The Fairness Index is a function of the variability of the throughput across the TCP connections defined as*

$$F((x_1, e_1), \ldots, (x_n, e_N)) = \frac{(\sum_{i=1}^{1=N} x_i/e_i)^2}{N \times \sum_{i=1}^{i=N} (x_i/e_i)^2}$$

where x_i = *observed throughput of the ith TCP connection* ($0 < i \leq N$),

and e_i = *expected throughput or fair share for connection* i.

For a symmetrical configuration using TCP over UBR, e_i can be defined as an equal share of the bottleneck link capacity ($e_i = C/N$). Thus, the fairness index metric applies well to N-source symmetrical configurations. In this case, note that when $x_1 = x_2 = \ldots = x_n$ then fairness index = 1. Also, low values of the fairness index represent poor fairness among the connections. The desired values of the fairness index must be close to 1. We consider a fairness index of 0.99 to be near perfect. A fairness index of 0.9 may or may not be acceptable depending on the application and the number of sources involved. Also note that the fairness index may not be a good metric for a small number of connections. Details on the fairness metric can be found in [19]. This fairness index has been used in several studies including [23]. In general, for a more complex configuration, the value of e_i can be derived from a rigorous formulation of a fairness definition that provides max-min fairness to the connections.

Due to space constraints, in this paper, we do not present extensive fairness results, but provide brief discussions of fairness when appropriate. In [14], we provide more comprehensive fairness results, and show that with sufficient buffers and a large number of TCP sources, good fairness values are achieved over UBR.

3 TCP over UBR+

Since TCP congestion control is inherently limited by the round trip time, long delay paths have significant effects on the performance of TCP over ATM. A large delay-bandwidth link must be utilized efficiently to be cost effective. In this section, we first present performance results results of TCP over UBR and its enhancements, with satellite delays.

3.1 Performance Results for Satellite Delays

All simulations use the N source configurations shown in Figure 1. All sources are identical and persistent TCP sources i.e., the sources always send a segment as long as it is permitted by the TCP window. Moreover, traffic is unidirectional so that only the sources send data. The destinations only send ACKs. The delayed acknowledgment timer is deactivated, i.e., the receiver sends an ACK as soon as it receives a segment. Each TCP is transported over a single VC. This enables the switch to perform per-TCP control using per-VC control (Selective Drop). When multiple TCPs are aggregrated over a single VC, per-TCP accounting cannot be performed, and the buffer management within a single VC becomes equivalent to EPD or RED. Aggregrated TCP VCs are further discussed in section 6.

We consider the following factors while performing our experiments

TCP mechanism. TCP Vanilla, Reno and SACK as described in section 2.

Round Trip Latency: GEO (550 ms) and LEO (30 ms). Our primary aim is to study the performance of large latency connections. The typical one-way latency from earth station to earth station for a single LEO (700 km altitude, 60 degree elevation angle) hop is about 5 ms [20]. The one-way latencies for multiple LEO hops can easily be up to 50 ms from earth station to earth station. GEO one-way latencies are typically 275 ms from earth station to earth station. For GEO's, the link between the two switches in Figure 1 is a satellite link with a one-way propagation delay of 275 ms. The links between the TCP sources and the switches are 1 km long. This results in a round trip propagation delay of about 550 ms. The LEO configuration is modeled as a an access uplink to the on board satellite switch, one or more intersatellite hops, and a downlink to the earth terminal. For the set of simulations presented in this section, a single intersatellite link is used, and each link has a propagation delay of 5 ms, resulting in an end to end round trip time of 30 ms.

Switch Buffer Size. The buffer sizes used in the switch are 200,000 cells and 600,000 cells for GEO and 12,000 and 36,000 cells for LEO. These buffer sizes reflect aproximate bandwidth-delay equivalents of 1 RTT and 3 RTTs respectively. Similar buffer sizes have been used in [26] for studying TCP performance over ABR, and it is interesting to assess the performance of UBR in such situations. The relation between buffer sizes and round trip times is further explored in 5.

Switch discard policy. We use two discard policies, Early Packet Discard (EPD) and Selective Drop (SD) as described in section 2.

Higher priority cross traffic and guaranteed rates. We introduce the effects of cross traffic in section 4.

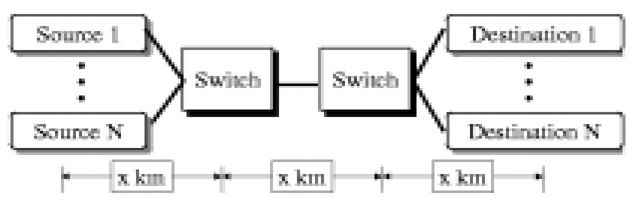

Figure 1: The N source TCP configuration

We first present the results for LEO and GEO systems with the following parameters:

- The number of sources (N) is set to 5. In general, the typical number of simultaneous sources might be active, but our simulations give a good representation of the ability of the TCPs to recover during congestion. In section 5 we further extend these results to a large number of

Table 1: TCP over Satellite (UBR): Efficiency

TCP Type	Buffer Size	Efficiency (LEO) EPD	SD	Efficiency (GEO) EPD	SD
SACK	1RTT	0.90	0.88	0.6	0.72
SACK	3RTT	0.97	0.99	0.99	0.99
Reno	1RTT	0.79	0.8	0.12	0.12
Reno	3RTT	0.75	0.77	0.19	0.22
Vanilla	1RTT	0.9	0.9	0.73	0.73
Vanilla	3RTT	0.81	0.81	0.81	0.82

sources.

- Cells from all TCP sources share a single FIFO queue in the each outgoing link.The FIFO is scheduled according to link availability based on the data rate. In these experiments, no other traffic is present in the network. Cross traffic is introduced in the next section.
- The maximum value of the TCP receiver window is 8,704,000 bytes for GEO and 600,000 bytes for LEO (the window scale factor is used). These window size are sufficient to fill the 155.52 Mbps pipe.
- The TCP maximum segment size is 9180 bytes. A large value is used because most TCP connections over ATM with satellite delays are expected to use large segment sizes.
- The duration of simulation is 40 seconds. This is enough time for the simulations to reach steady state.
- All link bandwidths are 155.52 Mbps.
- The effects of channel access such as DAMA are ignored in our simulations. This simplifies the analysis, and focuses on the properties of buffer management and end-system policies.

Table 1 shows the efficiency values for TCP over UBR with 5 TCP sources. The table lists the efficiency values for three TCP types, 2 buffer sizes, 2 drop policies and the 2 round trip times. Several conclusions can be made from the table:

Conclusion 1 (Performance of SACK) *For long delays, selective acknowledgments significantly* *improve the performance of TCP over UBR.* For sufficient buffers, the efficiency values are typically higher for SACK than for Reno and vanilla TCP. This is because SACK often prevents the need for a timeout and can recover quickly from multiple packet losses. Under severe congestion, SACK can perform worse than Vanilla. This is because under severe congestion, retransmitted packets can be dropped, and the SACK sender experiences a timeout. As a result, all SACK information is reneged and the sender starts with a congestion window of 1. The lower efficiency is due to the bandwidth wasted in the aggressive fast retransmission due to SACK. Reduced SACK performance under severe congestion has also been reported in [14].

Conclusion 2 (Performance of fast retransmit and recovery) *As delay increases, fast retransmit and recovery is detrimental to the performance of TCP.* The efficiency numbers for Reno TCP in table 1 are much lower than those of either SACK or Vanilla TCP. This is a well known problem with the fast retransmit and recovery algorithms in the presence of bursty packet loss. When multiple packets are lost during a single round trip time (the same window), TCP Reno reduces its window by half for each lost packet. The reduced window size is not large enough to send new packets that trigger duplicate acks, resulting in a timeout. The timeout occurs at a very low window (because of multiple decreases during fast recovery), and the congestion avoidance phase triggers at a low window. For large RTT, the increase in window during congestion avoidance is very inefficient, and results in much capacity being unused. For a large number of TCPs, the total throughput is greater, but this reflects a worst case scenario and highlights the inneficiency of the congestion avoidance phase for large round trip times. Vanilla TCP performs better, because the first packet loss triggers a timeout when the window is relatively large. The ensuing slow start phase quickly brings the window to half its original value before congestion avoidance sets in.

Conclusion 3 (Performance of buffer management) *The effect of intelligent buffer management policies studied above is not significant in satellite networks.* It has been shown that both EPD and Selective Drop improve the performance of TCP over UBR for terrestrial networks [16]. However, in these experiments, intelligent drop policies have little effect on the performance of TCP over UBR. The primary reason is that in our simulations, we have used adequate buffer sizes for high performance. Drop policies play a more significant role in improving performance in cases where buffers are a limited resource. These findings are further corroborated for WWW traffic by [14].

4 Effect of Higher Priority Traffic

In the presence of higher priority traffic sharing the satellite link, UBR traffic can be temporarily starved. This may have adverse effects on TCP depending on the bandwidth and duration of the higher priority traffic. Providing a guaranteed rate to UBR traffic has been suggested as a possible solution to prevent bandwidth starvation. The rate guarantee is provided to the entire UBR service category. Per-VC guarantees to UBR are not provided in this architecture. Such a minimum rate guarantee can be implemented using a simple scheduling mechanism like weighted round robin or weighted fair queuing.

To demonstrate the effect of VBR on TCP over UBR, we simulated a five source configuration with a 30 ms round trip time. An additional variable bit rate (VBR) source-destination pair was introduced in the configuration. The VBR traffic followed the same route as the TCP traffic, except that it was put into a separate queue at the output port in each traveresed switch. The VBR queue was given strict priority over the UBR queue, i.e., the UBR queue was serviced only if the VBR queue was empty. The VBR source behaved as an on-off source sending data at a constant rate during the on period, and not sending any data during the off period. On-off background sources have been used in several studies for TCP over UBR and ABR [26]. Three different VBR on/off periods were simulated – 300ms, 100ms and 50ms. In each case, the on times were equal to the off times and, during the on periods, the VBR usage was 100% of the link capacity. The overall VBR usage was thus 50% of the link capacity.

The effect of UBR starvation is seen in table 2. The table shows the efficiency in the presence of VBR traffic. Note that in calculating efficiency, the bandwidth used by the VBR source is taken into account. From the table we can see that longer VBR bursts (for the same average VBR usage of 50%) result in lower throughput for TCP over UBR. At 300 ms on-off times, the efficiency values

Table 2: LEO: SACK TCP with VBR (strict priority) : Efficiency

Buffer (cells)	VBR period (ms)	Efficiency EPD	SD
12000	300	0.43	0.61
36000	300	0.52	0.96
12000	100	0.58	0.70
36000	100	0.97	0.97
12000	50	0.65	0.73
36000	50	0.98	0.98

were very low, even for large buffer sizes. The reason for low throughput was TCP timeout due to starvation. When no TCP packets were sent for periods longer than the TCP RTO value, the source TCP times out and enters slow start. For large buffer sizes, the efficiency was better, because the packets were all queued during starvation.

We also performed simulations with the GEO configuration in the presence of VBR. The corresponding efficiencies for SACK TCP over GEO were much higher than those for LEO. The results (in table 7 in the appendix) show that SACK TCP over GEO achieves near optimal throughput even in the presence of bursty VBR traffic. The performance of Reno TCP was poor, and that corroborates the poor performance of Reno without the VBR sources. Longer periods of starvation (much more than the round trip times) do reduce throughput even in GEOs, but such starvation periods are unrealistic in high bandwidth links. Starvation due to satellite link outages can be of this duration, but this problem cannot be solved by providing rate guarantees, and its study is beyond the scope of this work.

To improve the performance of TCP over LEO delays, we have proposed the use of Guaranteed Rate (GR) for the UBR service category. Guaranteed Rate provides a minimum rate guarantee to the entire UBR service category. UBR cells are queued onto a single FIFO queue. The guarantee is provided using a service discipline like weighted round robin that reserves a least a minimum fraction of the link capacity for the UBR queue.

Figures 2,3,4, and 5 show the key results on effect of a minimum rate guarantee to UBR in the

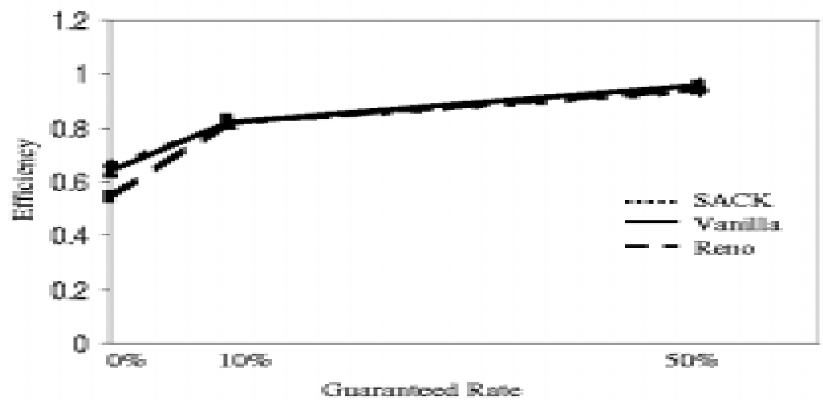

Figure 2: LEO: Guaranteed Rate vs TCP

presence of high priority Variable Bit Rate (VBR) traffic (Table 6 in the appendix lists the complete results in the figures). The VBR traffic is modeled as a simple on-off model with an on-off period of 300ms. The table shows the values of efficiency for 5 and 15 TCP sources and a single VBR source running over a satellite network.

The following parameter values were used for this experment:

- *Number of Sources*: 5 and 15.
- *Buffer size*: 1RTT and 3RTT (delay-bandwidth product).
- *TCP version*: Vanilla, Reno and SACK.
- *Switch Drop Policy*: Selective Drop and Early Packet Discard.
- *Guaranteed Rate*: 0%, 10% and 50% of the total link capacity.
- *Round trip latency*: 30 ms (LEO) and 550 ms (GEO)

In this experiment, we are mainly interested in the effects of TCP, Guaranteed Rate and buffer size. Also, preliminary analysis from the table has shown that the switch drop polices do not have a significant effect on performance. The relative change in efficiencies due to changes in the number of sources is also not significant. The key factors whose effects on efficiency are under study are Guaranteed Rate, buffer size and TCP version.

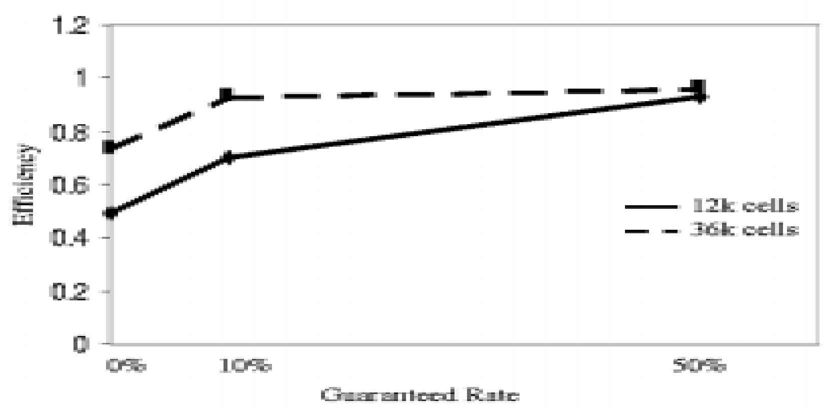

Figure 3: LEO: Guaranteed Rate vs Buffer Size

Figure 2 shows the relative effects of GR and TCP mechanisms on the efficiency for 30 ms RTT. Each point in the figure represents the efficiency value averaged over all the other factors above (Number of sources, Buffer Size and Switch Drop Policy). The figure illustrates that in the presence of high priority traffic, the effect of TCP for smaller round trip times is largely inconsequential. The key determinant is the amount of constant bandwidth allocated to the TCP traffic. The figure shows that even a 10% bandwidth reservation can increase the overall throughput by about 25%.

Figure 3 shows the relative effects of GR and buffer size mechanisms on LEO efficiency. Each point in the figure represents the efficiency value averaged over all the other factors (Number of sources, drop policy, TCP mechanism). The figure shows that a 10% GR allocation increases the efficiency by about 20%. A larger buffer size (36k cells) along with 10% GR can provide high efficiency.

Figures 4, and 5 illustrate the corresponding results for GEO delays. Both figures show that the effect of GR and buffer are incsignificant relative to the effect of TCP. Reno performs very poorly, while SACK performs the best.

The following conclusions can be drawn from the experiments so far:

Conclusion 4 (End system policies vs drop policies) *For longer delays, end system policies are more important than network based drop policies.* For short delays, drop policies have some

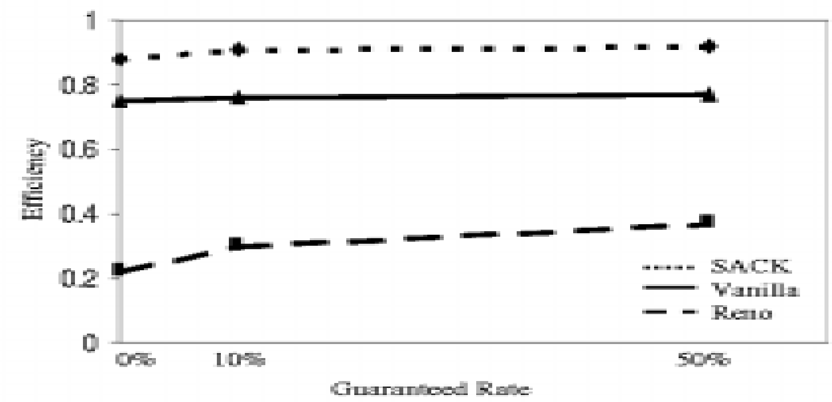

Figure 4: GEO: Guaranteed Rate vs TCP

effect. For long delays, TCP SACK provides the best performance among all TCP mechanisms studied.

Conclusion 5 (uaranteed rates vs end system policies) *Guaranteed rate helps in the presence of high priority VBR traffic.* The effect of guaranteed rate is more significant for shorter delays (LEO). For longer (GEO) delays, TCP SACK is the most important factor.

In the remainder of the paper, we will present results using TCP SACK. Although SACK is currently not widely deployed, it is quickly becoming the protocol of choice in many new implementations. Moreover, several satellite systems are considering the use of Performance Enhancing Proxies (PEP) [25] over the satellite segment. These proxies will invariably use SACK (or an improvement) on the satellite link, and it is interesting to assess performance using SACK as the desired TCP behavior.

5 Buffer Requirements for TCP over UBR

Our results have shown that small switch buffer sizes can result low TCP throughput over UBR. It is also clear, that the buffer requirements increase with increasing delay-bandwidth product of the connections (provided the TCP window can fill up the pipe). However, the studies have not quantitatively analyzed the effect of buffer sizes on performance. *As a result, it is not clear how*

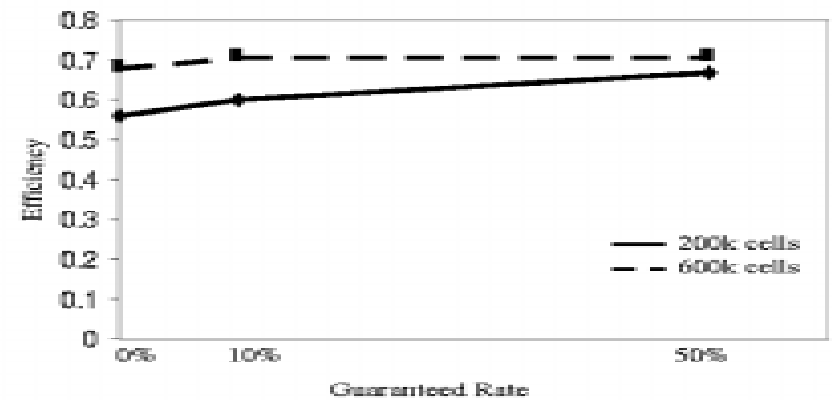

Figure 5: GEO: Guaranteed Rate vs Buffer Size

the increase in buffers affects throughput, and what buffer sizes provide the best cost-performance benefits for TCP/IP over UBR. In this section, we present simulation experiments to assess the buffer requirements for various satellite delay-bandwidth products for TCP/IP over UBR.

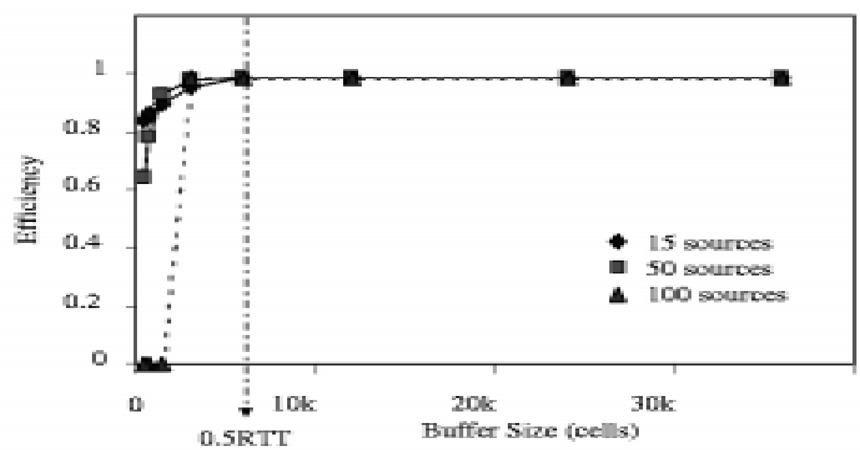

Figure 6: Buffer requirements for single hop LEO

5.1 Parameters

We study the effects of the following parameters:

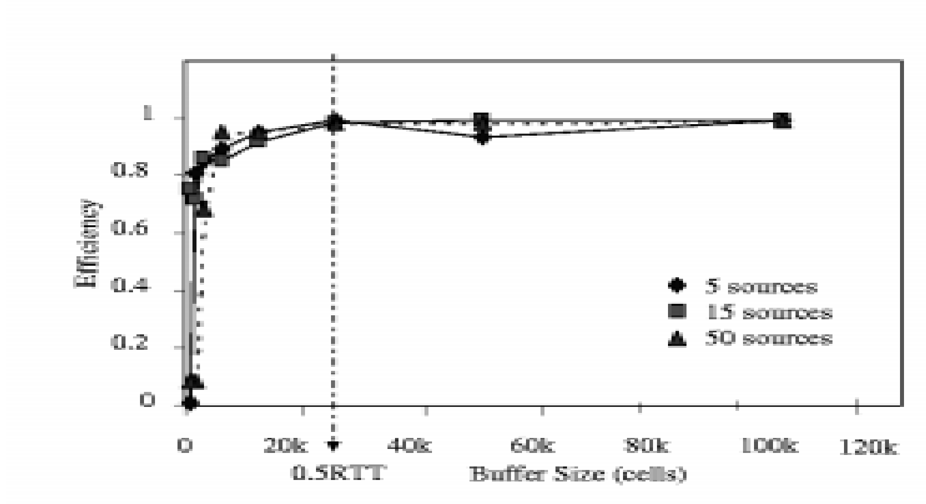

Figure 7: Buffer requirements for multiple hop LEO

Round trip latency. In addition GEO (550 ms round trip) and LEO (30 ms round trip), we also study a multi-hop LEO with an intersatellite one way delay of 50 ms. This results in a round trip time of 120 ms.

Number of sources. To ensure that the results are scalable and general with respect to the number of connections, we will use configurations with 5, 15 and 50 TCP connections on a single bottleneck link. For the single hop LEO configuration, we use 15, 50 and 100 sources.

Buffer size. This is the most important parameter of this study. The set of values chosen are $2^{-k} \times$ Round Trip Time (RTT), $k = -1..6$, (i.e., 2, 1, 0.5, 0.25, 0.125, 0.0625, 0.031, 0.016 multiples of the round trip delay-bandwidth product of the TCP connections.)

The buffer sizes (in cells) used in the switch are the following:

- *LEO (30 ms)*: 375, 750, 1500, 3 K, 6 K, 12 K (=1 RTT) , 24 K and 36 K.

  Multiple LEO (120 ms): 780, 1560, 3125, 6250, 12.5 K, 50 K (=1 RTT) , and 100 K.

- *GEO (550 ms)*: 3375, 6750, 12500, 25 K, 50 K, 100 K, 200 K (=1 RTT) , and 400 K.

Switch drop policy. We use the per-VC buffer allocation policy, Selective Drop (see [16]) to fairly allocate switch buffers to the competing connections.

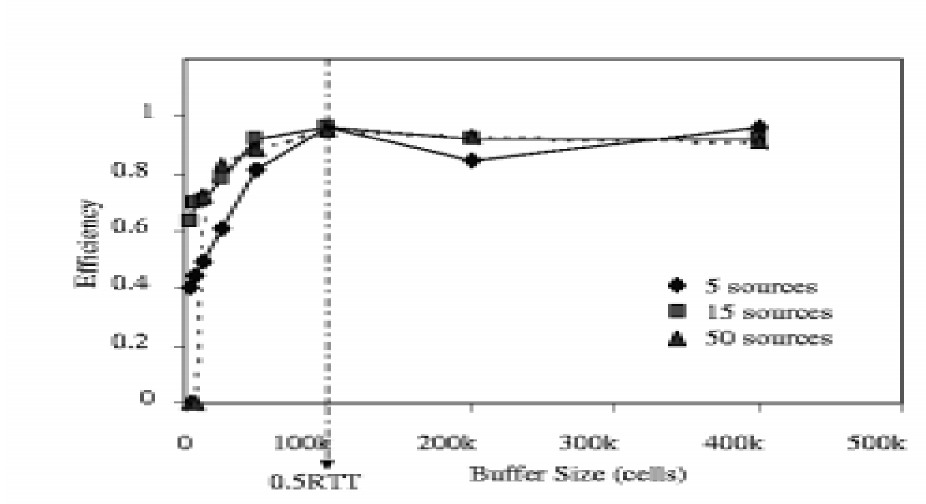

Figure 8: Buffer requirements for GEO

End system policies. We use SACK TCP for this study. Further details about our SACK TCP implementation can be found in [6].

The maximum value of the TCP receiver window is 600000 bytes, 2500000 bytes and 8704000 bytes for single hop LEO, multi-hop LEO and GEO respectively. These window sizes are sufficient to fill the 155.52 Mbps links. The TCP maximum segment size is 9180 bytes. The duration of simulation is 100 seconds for multi-hop LEO and GEO and 20 secs for single hop LEO configuration. These are enough for the simulations to reach steady state. All link bandwidths are 155.52 Mbps, and peak cell rate at the ATM layer is 149.7 Mbps after the SONET overhead.

We plot the buffer size against the achieved TCP throughput for different delay-bandwidth products and number of sources. The asymptotic nature of this graph provides information about the optimal buffer size for the best cost-performance ratio.

5.2 Simulation Results

Figures 6, 7 and 8 show the resulting TCP efficiencies for the 3 different latencies. Each point in the figure shows the efficiency (total achieved TCP throughput divided by maximum possible throughput) against the buffer size used. Each figure plots a different latency, and each set of points (connected by a line) in a figure represents a particular value of N (the number of sources).

For very small buffer sizes, (0.016×RTT, 0.031×RTT, 0.0625×RTT), the resulting TCP throughput is very low. In fact, for a large number of sources (N=50) , the throughput is sometimes close to zero. For moderate buffer sizes (less then 1 round trip delay-bandwidth), TCP throughput increases with increasing buffer sizes. TCP throughput asymptotically approaches the maximal value with further increase in buffer sizes. TCP performance over UBR for sufficiently large buffer sizes is scalable with respect to the number of TCP sources. The throughput is never 100%, but for buffers greater than 0.5×RTT, the average TCP throughput is over 98% irrespective of the number of sources. Fairness (not shown here) is high for a large number of sources. This shows that TCP sources with a good per-VC buffer allocation policy like selective drop, can effectively share the link bandwidth.

The knee of the buffer versus throughput graph is more pronounced for larger number of sources. For a large number of sources, TCP performance is very poor for small buffers, but jumps dramatically with sufficient buffering and then stays about the same. For smaller number of sources, the increase in throughput with increasing buffers is more gradual.

For large round trip delays, and a small number of sources, a buffer of 1 RTT or more can result in a slightly reduced throughput (see figures 7 and 8). This is because of the variability in the TCP retransmission timer value. When the round trip is of the order of the TCP timer granularity (100 ms in this experiment), and the queuing delay is also of the order of the round trip time, the retransmission timeout values become very variable. This may result in false timeouts and retransmissions thus reducing throughput.

Conclusion 6 (Buffer requirements for TCP over satellite) *The simulations show that a buffer size of 0.5RTT is sufficient to provide high efficiency and fairness to TCPs over UBR+ for satellite networks.*

6 The Guaranteed Frame Rate Service

The enhancements to TCP over UBR can provide high throughput to TCP connections over satellite networks. However, UBR does not provide any guarantees to its VCs. The service received by

UBR connection is implementation dependent. Service guarantees may be useful for a satellite-ATM network connecting multiple network clouds of Virtual Private Networks. It may be desirable to provide minimum rate guarantees to VCs of each VPN. Per-VC minimum rate guarantees can be implemented using either the Guaranteed Frame Rate (GFR) service or the Available Bit Rate (ABR) service. In this section we will describe how to implement per-VC minimum rate guarantees for the GFR service over satellite networks.

Guaranteed Frame Rate provides a minimum rate guarantee to VCs, and allows for the fair usage of any extra network bandwidth. GFR is a frame based service and uses AAL5 which enables frame boundaries to be visible at the ATM layer. The service requires the specification of a maximum frame size (MFS) of the VC. If the user sends packets (or frames) smaller than the maximum frame size, at a rate less than the minimum cell rate (MCR), then all the packets are expected to be delivered by the network with minimum loss. If the user sends packets at a rate higher than the MCR, it should still receive at least the minimum rate. A leaky bucket like mechanism called Frame-GCRA is used to determine fir a frame is eligible for MCR guarantees. Such frames are called QoS eligible. The minimum rate is guaranteed to the CLP=0 frames of the connection. In addition, a connection sending in excess of the minimum rate should receive a fair share of any unused network capacity. The exact specification of the fair share has been left unspecified by the ATM Forum.

GFR requires minimum signaling and connection management functions, and depends on the network's ability to provide a minimum rate to each VC. GFR is likely to be used by applications that can neither specify the traffic parameters needed for a VBR VC, nor have capability for ABR (for rate based feedback control). Current internetworking applications fall into this category, and are not designed to run over QoS based networks. These applications could benefit from a minimum rate guarantee by the network, along with an opportunity to fairly use any additional bandwidth left over from higher priority connections. The detailed GFR specification is provided in [1], but the above discussion captures the essence of the service.

6.1 GFR Implementation Options

There are three basic design options that can be used by the *network* to provide the per-VC minimum rate guarantees for GFR – tagging, buffer management, and queueing:

> *Tagging: Network based tagging* (or policing) can be used to mark non-eligible packets before they enter the network. Network based tagging on a per-VC level requires some per-VC state information to be maintained by the network and increases the complexity of the network element. Tagging can isolate eligible and non-eligible traffic of each VC so that other rate enforcing mechanisms can use this information to schedule the conforming traffic in preference to non-conforming traffic.

> *Buffer management:* Buffer management is typically performed by a network element to control the number of packets entering its buffers. In a shared buffer environment, where multiple VCs share common buffer space, per-VC accounting can control the buffer occupancies of individual VCs. Per-VC accounting introduces overhead, but without per-VC accounting it is difficult to control the buffer occupancies of individual VCs (unless non-conforming packets are dropped at the entrance to the network by the policer). Note that per-VC buffer management uses a single FIFO queue for all the VCs. This is different from per-VC queuing and scheduling discussed below.

> *Scheduling:* While tagging and buffer management control the entry of packets into a network element, queuing strategies determine how packets are scheduled onto the next hop. Per-VC queuing maintains a separate queue for each VC in the buffer. A scheduling mechanism can select between the queues at each scheduling time. However, scheduling adds the cost of per-VC queuing and the service discipline. For a simple service like GFR, this additional cost may be undesirable.

A desirable implementation of GFR is to use a single queue for all GFR VCs, and provide minimum rate guarantees by means of intelligent buffer management policies on the FIFO. Several proposals have been made [2, 3, 4] to provide rate guarantees to TCP sources with FIFO queuing in the network. The bursty nature of TCP traffic makes it difficult to provide per-VC rate guarantees using FIFO queuing. In these proposals, per-VC scheduling was recommended to provide rate guarantees to TCP connections. However, all these studies were performed at high target network utilization, i.e., most of the network capacity was allocated to the MCRs. The designers of the GFR service have intended to allocate MCRs conservatively. Moreover, these proposals are very aggressive in dropping TCP packets causing TCP to timeout and lose throughput. All the above studies have examined TCP traffic with a single TCP per VC. However, routers that use GFR VCs, will multiplex many TCP connections over a single VC. For VCs with several aggregated TCPs, per-VC control is unaware of each TCP in the VC. Moreover, aggregate TCP traffic characteristics and control requirements may be different from those of single TCP streams.

In the next subsection, we will briefly describe a buffer management policy called Differential Fair Buffer Allocation (DFBA) that provides per-VC minimum rate guarantees. We present the performance of DFBA for LEO and GEO systems. A complete analysis of DFBA for terrestrial networks is presented in [21].

6.2 The Differential Fair Buffer Allocation Scheme

The Differential Fair Buffer Allocation (DFBA) scheme is based on per-VC accounting on a FIFO buffer. The scheme maintains efficiency and fairness in the network by selectively accepting or discarding incoming cells of a VC. Once the cells are queued, they are serviced in a FIFO manner from the GFR queue. DFBA recognizes frame boundaries using the EOM bit in the last cell of a frame. As a result, DFBA is fully compliant with the GFR requirements specified by the ATM forum.

DFBA uses the current queue length (buffer occupancy) as an indicator of network load. The scheme tries to maintain an optimal load so that the network is efficiently utilized, yet not congested. Figure 9 illustrates the operating region for DFBA. The high threshold (H) and the low threshold (L) represent the cliff and the knee respectively of the classical load versus delay/throughput graph. The goal is to operate between the knee and the cliff.

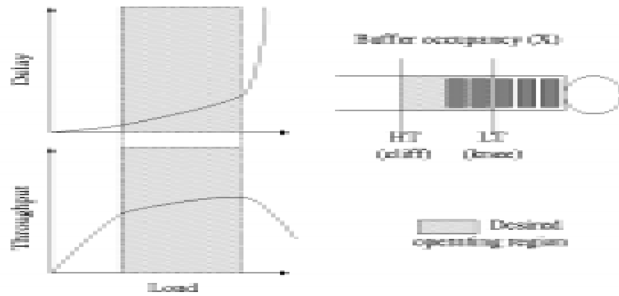

Figure 9: DFBA Target Operating Region

In addition to efficient network utilization, DFBA is designed to allocate buffer capacity fairly amongst competing VCs. This allocation is proportional to the MCRs of the respective VCs. The following variables are used by DFBA to fairly allocate buffer space:

X = Total buffer occupancy at any given time

L = Low buffer threshold

H = High buffer threshold

MCR_i = MCR guaranteed to VC_i

W_i = Weight of VC_i = MCR_i/(GFR capacity)

$W = \Sigma W_i$

X_i = Per-VC buffer occupancy ($X = \Sigma X_i$)

Z_i = Parameter ($0 \leq Z_i \leq 1$)

DFBA maintains the total buffer occupancy (X) between L and H. When X falls below L, the scheme attempts to bring the system to efficient utilization by accepting all incoming packets. When X rises above H, the scheme tries to control congestion by performing EPD. When X is between L and H, DFBA attempts to allocate buffer space in proportional to the MCRs, as determined by the W_i for each VC. When X is between L and H, the scheme also drops low priority (CLP=1) packets so as to ensure that sufficient buffer occupancy is available for CLP=0 packets.

Figure 10 illustrates the four operating regions of DFBA. The graph shows a plot of the current buffer occupancy X versus the normalized fair buffer occupancy ($\bar{X}_i$) for VC_i. If VC_i has a weight W_i, then its target buffer occupancy (X_i) should be $X \times W_i/W$. Thus, the normalized buffer occupancy of VC_i can be defined as $\bar{X}_i = X_i \times W/W_i$. The goal is to keep $\bar{X}_i$ as close to X as possible, as indicated by the solid $y = x$ line in the graph. Region 1 is the underload region, in which the current buffer occupancy is less than the low threshold L. In this case, the scheme tries to improve efficiency. Region 2 is the region with mild congestion because X is above L. As a result, any incoming packets with CLP=1 are dropped. Region 2 also indicates that VC_i has a larger buffer occupancy than its fair share (since $X_i > X \times W_i/W$). As a result, in this region, the scheme drops some incoming CLP=0 packets of VC_i, as an indication to the VC that it is using more than its fair share. In region 3, there is mild congestion, but VC_i's buffer occupancy is below its fair share. As a result, only CLP=1 packets of a VC are dropped when the VC is in region 3. Finally, region 4 indicates severe congestion, and EPD is performed here.

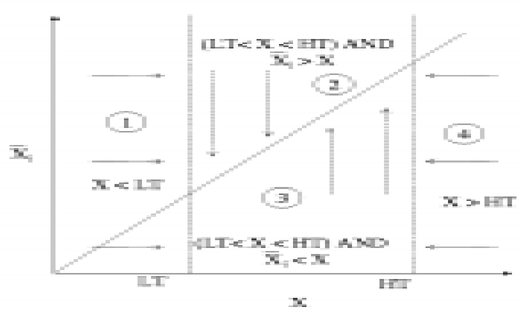

Figure 10: DFBA Drop Regions

In region 2, the packets of VC_i are dropped in a probabilistic manner. This drop behavior is controlled by the drop probability function P{drop}. This is further discussed below.

The probability for dropping packets from a VC when it is in region 2 can be based on several factors. Probabilistic drop is used by several schemes including RED and FRED. The purpose of probabilistic drop is to notify TCP of congestion so that TCP backs off without a timeout. An aggressive drop policy will result in a TCP timeout. Different drop probability functions have different effects on TCP behavior. In general, a simple probability function can use RED like drop, while a more complex function can depend on all the variables defined above. The drop probability used in our simulations is described in detail in [21] and is given by:

$$P\{drop\} = Z_i \times (\alpha \times \frac{X_i - X \times W_i/W}{X \times (1 - W_i/W)} + (1-\alpha)\frac{X-L}{H-L})$$

For satellite latencies, an important parameter in this equation is Z_i. It has been shown [22] that for a given TCP connection, a higher packet loss rate results in a lower average TCP window. As a result, a higher drop probability also results in a lower TCP window. In fact, it has been shown [22], that for random packet loss, the average TCP window size is inversely proportional to the square root of the packet loss probability. As a result, the average TCP data rate D is given by

$$D \propto \frac{MSS}{RTT\sqrt{P\{\mathrm{drop}\}}}$$

The data rate is in fact determined by the window size and the RTT of the connection. To maintain a high data rate, the desired window size should be large. As a result, the drop probability should be small. Similarly when the RTT is large, a larger window is needed to support the same data rate (since the delay-bandwidth product increases). As a result, a smaller drop rate should be used. DFBA can be tuned to choose a small Z_i for large latency VCs, as in the case of switches connected to satellite hops, or for VCs with high MCRs. The inherent limitation of any buffer management scheme that depends only on local state is seen here. In general, the switch does not know the RTT of a VC. The switch must estimate a connection's RTT using local state such as the propagation delay of its outgoing links. In case of satellite switches, this propagation delay is likely to be the dominant delay in the VCs path. As a result, the local state provides a pretty good estimate of the today delay. Terrestrial switches are limited in this respect. This limititation is also discussed in [23].

Another potential limitation of any such scheme is that the granularity of fairness is limited by the granularity of flows. The fairness is guaranteed between VCs but not within the TCPs of each. This limitation is not only peculiar to ATM but also to IP. IP routers typically define flows according to IP address or network address source-destination pairs. TCP/UDP port level granularities are not a scalable solution for backbone networks. As as result, the TCP connections within an IP flow suffer the same kind of unfairness as TCP connections within ATM VCs. However, the probabilistic drop randomizes the packets dropped within a VC. Thus, the scheme can maintain RED like fairness among the TCPs within a VC. This can be accomplished by using a RED like drop probability for drop.

6.3 Simulation Results

The test results presented here are with DFBA for Satellite-ATM interconnected TCP/IP networks. Figure 11 illustrates the basic test configuration. The figure shows 5 local IP/ATM edge switches connected to backbone ATM switches that implement GFR. Each local switch carries traffic from multiple TCPs as shown in the figure. The backbone link carries 5 GFR VCs, one from each local network. Each VC thus carries traffic from several TCP connections. We used 20 TCPs per VC for a total of 100 TCPs. The GFR capacity was fixed to the link rate of 155.52 Mbps (approx. 353207 cells per sec). The MCRs were 20, 40, 60, 80 and 100 kcells/sec for VCs 1...5 respectively, giving a total MCR allocation of 85% of the GFR capacity. At the TCP layer, these MCR's translated to expected TCP throughputs of 6.91, 13.82, 20.74, 27.65, 34.56 Mbps respectively. Note that, in GFR deployments, MCRs are expected to be allocated more conservatively, and 85% allocation reflects an upper bound on MCR allocation. Also, these numbers are aggregate numbers for all 20 TCPs for VCs 1 through 5. All TCP sources are persistent TCPs with SACK. Based on previous studies, [5], we set the thresholds L and H to 0.5 and 0.9 of the buffer capacity respectively. A complete parameter study of DFBA is presented in [21].

In figure 11, the access hop is denoted by x, and the backbone hop is denoted by y. Three different simulation configurations are presented below:

WAN with homogeneous RTT. We first present DFBA results with one way backbone delay = 5 ms, and negligible access delay. In this case, three different buffer sizes were simulated in the bottleneck backbone switch – 25000, 6000 and 3000 cells. The goal of this experiment is to illustrate that DFBA achieves the unequal MCR guarantees for each VC. Table 3 lists the expected and achieved throughputs for each VC in the configuration. The achieved

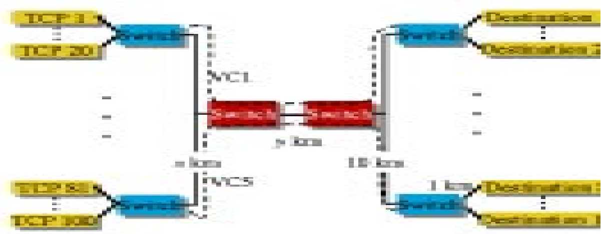

Figure 11: DFBA Simulation Confuration

throughput for a VC is the sum of all the TCP throughputs in that VC. The table illustrates that for each of the buffer sizes, the achieved throughputs exceed the expected throughputs for all VCs. As a result, DFBA provides MCR guarantees to aggregated TCP traffic. The overall efficiency of the system is also more than 95% resulting in high network utilization. In the simulations, the excess capacity (GFR capacity - MCR allocation) is almost equally distributed among the five VCs. This allocation may or may not be considered fair because VCs with higher MCRs may demand a higher portion of the excess. [21] discusses techniques to provide MCR proportional allocation of excess capacity using DFBA.

LEO Access with heterogenous RTT. In this configuration, the access hop (x) for VC 3, is a LEO link with a 25 ms one way delay. This results in a round trip delay of 60 ms for VC3. All other VCs still have negligible access delay, and the backbone delay is also 5 ms one way. The results of this simulation with buffer size = 6000 cells is shown in table 4. The table again shows that DFBA provides the allocated rates to VCs with different MCRs.

GEO backbone Finally, we present the case where the backbone hop is a GEO link. The round trip delay in this case is abount 550 ms. The GEO hop is the most dominant hop with respect to latency, and thus, in the simulation the access hops had negligible latency. Figure 5 shows the achieved throughputs for three different buffer sizes. Again, the table shows that DFBA provides MCR guarantees to VCs over long delay networks.

The ideas and results from this section can be summarized as follows:

Table 3: Minimum rate guarantees with DFBA.

Expected	Achieved Throughput (Mbps)		
Throughput (Mbps)	25k buffer	6K buffer	3k buffer
6.91	11.29	11.79	10.02
13.82	18.19	18.55	19.32
20.74	26.00	25.13	25.78
27.65	32.35	32.23	32.96
34.56	39.09	38.97	38.56

Table 4: Minimum rate guarantees with DFBA. VC3 = LEO access

Expected Throughput (Mbps)	Achieved Throughput (Mbps)
6.91	10.55
13.82	17.06
20.74	24.22
27.65	33.74
34.56	41.10

Table 5: Minimum rate guarantees with DFBA. GEO backbone

Expected	Achieved Throughput		(Mbps)
Throughput (Mbps)	200k buffer	150K buffer	100k buffer
6.91	12.4	12.8	11.4
13.82	14.96	16.17	16.99
20.74	21.86	21.63	24.56
27.65	32.10	30.25	33.72
34.56	40.21	39.84	35.52

Conclusion 7 (FR Service) *The Guaranteed Frame Rate service is designed for frame based best effort applications, and supports per-VC minimum cell rate guarantees.*

Conclusion 8 (FR Implementation Options) *GFR can be implemented using tagging, buffer management and per-VC scheduling. A desirable implementation of GFR is by using a FIFO buffer with intelligent buffer management.*

Conclusion 9 (DFBA Results) *The Differential Fair Buffer Allocation (DFBA) scheme is a FIFO scheme that provides per-VC MCR guarantees to VCs carrying TCP traffic. Simulations with DFBA show that DFBA can provide such guarantees for terrestrial as well as satellite latencies.*

Conclusion 10 (Limitations) *In general, buffer management schemes for TCP/IP are limited by TCPs dependency on RTT, and the granularity of IP or ATM flows.*

7 Summary of Results

This paper describes a set of techniques for improving the performance of TCP/IP over Asynchronous Transfer Mode (ATM) based satellite networks. Among the service categories provided by ATM networks, the most commonly used category for data traffic is the unspecified bit rate (UBR) service. UBR allows sources to send data into the network without any network guarantees or control.

Several issues arise in optimizing the performance of Transmission Control Protocol (TCP) when ATM-UBR service is used over satellite links. In this paper, we studied several TCP mechanisms as well as ATM-UBR mechanisms to improve TCP performance over long-delay ATM networks. The UBR mechanisms that we studied in this project are:

- UBR with frame level discard policies,
- UBR with intelligent buffer management,
- UBR with guaranteed rate,
- Guaranteed Frame Rate (GFR).

The following TCP mechanisms were studied:

- Vanilla TCP with slow start and congestion avoidance,
- TCP Reno with fast retransmit and recovery,
- TCP with selective acknowledgements (SACK)

We studied several combinations of these mechanisms using an extensive set of simulations and quantified the effect of each of these mechanisms. The following summarizes the list of conclusions drawn from our simulations:

1. In several cases, Vanilla TCP over the UBR service category achieves low throughput and low fairness over satellite networks. This is because during packet loss, TCP loses significant amount of time waiting for retransmission timeout.

2. In the presence of bursty packet losses, fast retransmit and recovery (FRR) (without SACK) further hurts TCP performance over UBR for long delay-bandwidth product networks.

3. Frame level discard policies such as early packet discard (EPD) improve the throughput over cell-level discard policies. However, the fairness is not guaranteed unless intelligent buffer management with per virtual circuit (VC) accounting is used.

4. Throughput increases further with more aggressive New Reno and SACK. SACK gives the best performance in terms of throughput. We found that for long delay paths, the throughput improvement due to SACK is more than that from discard policies and buffer management.

5. A buffer size equal to about half the round-trip delay-bandswidth product of the TCP connections was found to be sufficient for high TCP throughput over satellite-UBR.

6. The presence of bursty high priority cross traffic can degrade the performance of TCP over UBR for terrestrial and low delay satellite networks. The effect of cross traffic is not very significant for GEO because the starvation time is relatively small compared to the round trip time for GEOs

7. Providing guaranteed rate to UBR helps in the presence of a high load of higher priority traffic. We found that reserving just a small fraction, say 10For GEO sytems, the effect of TCP SACK was more significant than other factors.

8. The GFR service category can provide per-VC MCR guarantees. The Differential Fair Buffer Allocation (DFBA) scheme provides MCR guarantees to GFR with a single queue using only per-VC accounting.

The results described above have been based on simulations using persistent TCP traffic. In [14], we have shown that the results also hold for world-wide web TCP traffic.

References

[1] ATM Forum, "ATM Traffic Management Specification Version 4.1," December 1998.

[2] Debashis Basak, Surya Pappu, "GFR Implementation Alternatives with Fair Buffer Allocation Schemes," ATM Forum 97-0528.

[3] Olivier Bonaventure, "A simulation study of TCP with the proposed GFR service category," DAGSTUHL Seminar 9725, High Performance Networks for Multimedia Applications, June 1997, Germany

[4] R. Goyal, R. Jain et.al, "Simulation Experiments with Guaranteed Frame Rate for TCP/IP Traffic," ATM Forum 97-0607.

[5] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy and Seong-Cheol Kim, "UBR+: Improving Performance of TCP over ATM-UBR Service," Proc. ICC'97, June 1997.

[6] R. Goyal, R. Jain et.al., "Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks," To appear, Proc. IC3N'97, September 1997. [1]

[1] All our papers and ATM Forum contributions are available from http://www.cis.ohio-state.edu/~jain

[7] Sastri Kota, Rohit Goyal, Raj Jain, "Satellite ATM Network Architectural Considerations and TCP/IP Performance", Proceedings of the 3rd Ka Band Utilization Converence, Italy, September 15-18, 1997, pp. 481-488, http://www.cis.ohio-state.edu/ jain/papers/kaband.htm

[8] Roch Guerin, and Juha Heinanen, "UBR+ Service Category Definition," ATM FORUM 96-1598, December 1996.

[9] Roch Guerin, and Juha Heinanen, "UBR+ Enhancements," ATM FORUM 97-0015, February 1997.

[10] Allyn Romanov, Sally Floyd, "Dynamics of TCP Traffic over ATM Networks," IEEE Journal of Selected Areas In Telecommunications, May 1995.

[11] Dong Lin, Robert Morris, "Dynamics of Random Early Detection," Proceedings of SIGCOMM97, 1997.

[12] Sally Floyd, Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transaction on Networking, August 1993.

[13] C. Ponzoni, "High Data Rate On-board Switching," 3rd Ka-band Utilization Conference, September 1997.

[14] Mukul Goyal, Rohit Goyal, Raj Jain, Bobby Vandalore, Sonia Fahmy, Tom VonDeak, Kul Bhasin, Norm Butts, and Sastri Kota, "Performance Analysis of TCP Enhancements for WWW Traffic using UBR+ with Limited Buffers over Satellite Links", ATM Forum/98-0876R1, December 1998, http://www.cis.ohio-state.edu/ jain/atmf/a98-0876.htm

[15] H. Li, K.Y. Siu, H.T. Tzeng, C. Ikeda and H. Suzuki "TCP over ABR and UBR Services in ATM," Proc. IPCCC'96, March 1996.

[16] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, "Improving the Performance of TCP over the ATM-UBR service", Computer Communications, Vol. 21, No. 10, July 1998, pp. 898-911, http://www.cis.ohio-state.edu/ jain/papers/cc.htm

[17] Rohit Goyal, Xiangrong Cai, Raj Jain, Sonia Fahmy, Bobby Vandalore " Per-VC Rate Allocation Techniques for ATM-ABR Virtual Source Virtual Destination Networks," Proceedings of Globecom '98, Australia, November 1998, http://www.cis.ohio-state.edu/ jain/papers/globecom98.htm

[18] Mark Allman, editor, "Ongoing TCP Research Related to Satellites," IETF Draft, Work in Progress.

[19] Raj Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Simulation, and Modeling*, Wiley-Interscience, New York, NY April 1991.

[20] Satellite altitudes taken from "Lloyd's satellite constellation," http://www.ee.surrey.ac.uk/Personal/L.Wood/constellations/overview.html

[21] Rohit Goyal, Raj Jain, Sonia Fahmy, Bobby Vandalore, "Buffer Management for the GFR Service," ATM Forum/98-0405, July 1998, http://www.cis.ohio-state.edu/ jain/atmf/a98-0405.htm

[22] M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm",Computer Communication Review, volume 27, number 3, pp. 67-82, July 1997.

[23] Floyd, S., "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic," Computer Communication Review, Vol.21, No.5, October 1991, p. 30-47.

[24] http://www-nrg.ee.lbl.gov/floyd/epd.html

[25] http://tcppep.lerc.nasa.gov/tcppep/

[26] Shivkumar Kalyanaraman, "Traffic Management for the Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks," PhD Dissertation, The Ohio State University, 1997, xxiv + 429.

Table 6: LEO: TCP with VBR (300ms on/off) over UBR+ with GR : Efficiency

TCP	Buffer	GR	Selective Drop	EPD	
5	12000	SACK	0.5	0.93	0.94
5	12000	SACK	0.1	0.66	0.69
5	12000	SACK	0.0	0.43	0.61
5	36000	SACK	0.5	0.99	0.99
5	36000	SACK	0.1	0.98	0.96
5	36000	SACK	0.0	0.52	0.96
15	12000	SACK	0.5	0.85	0.90
15	12000	SACK	0.1	0.61	0.76
15	12000	SACK	0.0	0.48	0.58
15	36000	SACK	0.5	0.95	0.97
15	36000	SACK	0.1	0.94	0.97
15	36000	SACK	0.0	0.72	0.95
5	12000	Reno	0.5	0.96	0.94
5	12000	Reno	0.1	0.79	0.71
5	12000	Reno	0.0	0.45	0.33
5	36000	Reno	0.5	0.97	0.93
5	36000	Reno	0.1	0.96	0.75
5	36000	Reno	0.0	0.92	0.33
15	12000	Reno	0.5	0.94	0.97
15	12000	Reno	0.1	0.66	0.79
15	12000	Reno	0.0	0.53	0.51
15	36000	Reno	0.5	0.97	0.98
15	36000	Reno	0.1	0.96	0.97
15	36000	Reno	0.0	0.66	0.59
5	12000	Vanilla	0.5	0.97	0.96
5	12000	Vanilla	0.1	0.70	0.69
5	12000	Vanilla	0.0	0.36	0.42
5	36000	Vanilla	0.5	0.97	0.97
5	36000	Vanilla	0.1	0.90	0.94
5	36000	Vanilla	0.0	0.33	0.92
15	12000	Vanilla	0.5	0.92	0.96
15	12000	Vanilla	0.1	0.66	0.74
15	12000	Vanilla	0.0	0.61	0.67
15	36000	Vanilla	0.5	0.97	0.97
15	36000	Vanilla	0.1	0.96	0.97
15	36000	Vanilla	0.0	0.93	0.93

Table 7: GEO: TCP with VBR (300ms on/off) over UBR+ with GR

TCP	Buffer	GR	Selective Drop	EPD
SACK	200000	0.5	0.87	0.84
SACK	200000	0.1	0.78	0.88
SACK	200000	0.0	0.74	0.82
SACK	600000	0.5	0.99	0.99
SACK	600000	0.1	0.99	0.99
SACK	600000	0.0	0.99	0.99
Reno	200000	0.5	0.33	0.46
Reno	200000	0.1	0.24	0.26
Reno	200000	0.0	0.16	0.17
Reno	600000	0.5	0.35	0.36
Reno	600000	0.1	0.39	0.34
Reno	600000	0.0	0.30	0.28
Vanilla	200000	0.5	0.83	0.71
Vanilla	200000	0.1	0.71	0.76
Vanilla	200000	0.0	0.81	0.68
Vanilla	600000	0.5	0.79	0.78
Vanilla	600000	0.1	0.80	0.80
Vanilla	600000	0.0	0.76	0.77