INTERNATIONAL JOURNAL OF SATELLITE COMMUNICATIONS
Int. J. Satell. Commun. 2001; **19**:111–139

Buffer management and rate guarantees for TCP over satellite-ATM networks‡

Rohit Goyal[1,*,†], Raj Jain[2], Mukul Goyal[3], Sonia Fahmy[4], Bobby Vandalore[5], Sastri Kota[6], Norm Butts[7] and Thomas vonDeak[8]

[1]*Principal Software Engineer, Axiowave Networks, 100 Nickerson Road, Marlborough, MA 01752, U.S.A.*
[2]*Nayna Networks Inc., 157 Topaz St., Milpitas, CA 95035, U.S.A.*
[3]*AT & T Labs-Research, 180 Park Ave, Florham Park, NJ 07932, U.S.A.*
[4]*Purdue University, 1398 Computer Science Building, West Lafayette, IN 47907-1398, U.S.A.*
[5]*Department of Computer and Information Science, The Ohio State University, 395 Dreese Lab, 2015 Neil Avenue, Columbus, OH 43210-1277, U.S.A.*
[6]*Chief Network Architect, BMS, 3825 Fabian Way M/S: Z-59, Palo Alto, CA 94303, U.S.A.*
[7]*Lockheed Martin Mission Systems, Sunnyvale, CA 94089, U.S.A.*
[8]*NASA Lewis Research Center, MS: 54-6, 21000 Brookpark Road, Cleveland, OH 44135, U.S.A.*

Raj Jain is now at Washington University in Saint Louis, jain@cse.wustl.edu http://www.cse.wustl.edu/~jain/

SUMMARY

Future broadband satellite networks will support a variety of service types. Many such systems are being design with ATM or ATM-like technology. A majority of Internet applications use TCP for data transfer. As a result, these systems must efficiently transport TCP traffic and provide service guarantees to such traffic. Several mechanisms have been presented in recent literature to improve TCP performance. Most of these can be categorized as either TCP enhancements or network-based buffer management techniques. Providing minimum rate guarantees to TCP traffic has also been suggested as a way to improve its performance in the presence of higher priority traffic sharing the link. However, the relative performance of the TCP enhancements versus the buffer management schemes has not been analyzed for long latency networks. In this paper, we address three issues. First, we present a performance analysis of TCP over satellite-ATM links using a best effort service—the ATM unspecified bit rate (UBR) service. This analysis shows that the relative impacts of buffer management, TCP policies and rate guarantees on TCP performance, depend heavily on the latency of the network. Second, we show through simulations that the buffer size required in the network for high TCP performance is proportional to the delay-bandwidth product of the network. Third, we propose a buffer management scheme called differential fair buffer allocation (DFBA) and show how it is used to implement a service that provides minimum rate guarantees to TCP traffic. An example of such a service is the ATM guaranteed frame rate (GFR) service, which is being standardized by the ATM Forum and the ITU. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: buffer management; rate guarantees; TCP; ATM

* Correspondence to: Rohit Goyal, Principal Software Engineer, Axiowave Networks, 100 Nickerson Road, Marlborough, MA 01752, U.S.A.
† E-mail: rgoyal@axiowave.com
‡ Some results in Section 3.3 have appeared in Reference [1]. Results in Section 4 have appeared in Reference [2]. This paper is a much enhanced and consolidated version and no results have been published in a journal.

Contract/grant sponsor: NASA Lewis Research Center; contract/grant number: NA53-97198

1. INTRODUCTION

The TCP over satellite [1] working group in the IETF has designed mechanisms for enhancing TCP performance over satellite networks. The group has focused its efforts on modifying the TCP protocol to improve its performance over long-delay satellite links [2]. The research on TCP has primarily considered a best effort service framework. Recent developments in broadband communications have promoted the design of multiservice network architectures that will provide minimum rate guarantees to TCP traffic. The implementation of such architectures requires network-based traffic management techniques such as buffer management and scheduling to support QoS guarantees.

More than 50 per cent of the planned Ka-band satellite systems propose to use on-board ATM or ATM-like fast packet switching [3]. ATM based switching and processing provides a new set of techniques for traffic management in satellite networks. In particular, several buffer management schemes have been proposed to improve *TCP performance* over ATM networks [4–6]. For bent pipe and regenerative satellite systems, ground stations can also benefit from these buffer management techniques to improve end-to-end TCP performance.

In addition to providing best effort services such as the ATM unspecified bit rate (UBR) service, satellite networks must also support services that provide *minimum rate guarantees* to their subscribers' traffic. The ATM guaranteed frame rate (GFR) service is one such service that is being standardized by the ATM Forum and the ITU. Buffer management can be used in the network to implement the GFR service and provide rate guarantees to TCP traffic.

While buffer management techniques provide clear performance improvements for TCP over terrestrial networks [4,7], it is not clear if their benefits are substantial over satellite networks that have larger propagation delays. Also, buffer management mechanisms increase the complexity and hence the cost of designing on-board and ground-based network elements. As a result, the satellite network architect is faced with the complex decision of designing earth terminals and on-board switches for optimizing the cost-performance tradeoff.

In this paper, we study buffer management techniques in satellite networks for TCP transport. We present simulations for the various TCP and ATM enhancements and discuss their relative effects. Based on the experimental results and analysis, we provide guidelines for designing satellite-ATM network architectures that can efficiently transport TCP data. We address the following three problems in this paper:

Problem 1 (Performance analysis of TCP over satellite-UBR)

The goal is to study the effect of delay on TCP performance. We study the relative effects of three TCP policies, three buffer management policies and rate guarantees on TCP performance over the ATM UBR service.

The TCP policies are slow start and congestion avoidance (TCP Vanilla) [8], fast retransmit and recovery (TCP Reno) [9], and selective acknowledgments (TCP Sack) [10]. The buffer management policies are tail drop, early packet, discard (EPD) [6], and selective drop (SD) [4]. We also discuss the effect of providing a minimum rate guarantee to the entire UBR service category.

Problem 2 (Buffer requirements for TCP over satellite-UBR)

We present simulation results to calculate the switch buffer sizes that provide high TCP performance over satellite.

Problem 3 (Buffer management for Guaranteed Frame Rate over satellite)
We describe the GFR service category and propose the differential fair buffer allocation (DFBA) scheme. DFBA is designed for TCP traffic and uses a FIFO buffer to provide minimum rate guarantees to ATM VCs carrying TCP traffic.

The paper does not propose any new TCP enhancements, but analyses the performance of existing and proposed TCP mechanisms including TCP Vanilla, TCP Reno and TCP SACK. In Reference [11], we present a study on TCP New Reno. In this paper, we also propose a buffer management technique for high throughput, fairness and minimum rate guarantees to TCP traffic over satellite-ATM networks. The simulation and analysis are performed for various satellite latencies covering 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 Reference [2]. The emphasis on network design issues for traffic management and basic service guarantees for TCP over satellite-ATM is the unique contribution of this research.

2. DESIGN OPTIONS FOR TCP OVER SATELLITE-ATM

In this section we describe the design options for transporting TCP over satellite-ATM.

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

- *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 because packets are lost due to congestion. Two reasons for the poor performance are the coarse-grained TCP transmission timeout and TCP synchronization [12]. The performance of TCP over UBR can be enhanced in the following ways (UBR with one or more enhancements has been informally called UBR +):
  - *UBR with frame based discard like EPD*. Among frame-based discard policies, the early packet discard [6] policy is widely used [13]. EPD maintains a threshold R in the switch buffer. When the buffer occupancy exceeds R, all new incoming packets are dropped. Partially received packets are accepted if possible. In terrestrial networks, EPD improves the efficiency of TCP over UBR but does not improve fairness [7]. The effect of EPD on satellite latencies has not been exhaustively studied.
  - *UBR with intelligent buffer management*. The selective drop (SD) [4] 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 (X) exceeds $R \times K$, new packets are dropped depending on the VC_i's buffer occupancy (Y_i). In SD, a VC's *entire packets* is dropped if

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

where N_a is the number of active VCs (VCs with at least one cell in the buffer) and Z is a parameter ($0 < Z \leqslant 1$) used to scale the effective drop threshold. In terrestrial networks, SD has been shown to improve the fairness TCP connections running over UBR [4].

— *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. Preliminary simulations [14] have shown that higher priority traffic can degrade TCP performance in some cases. In this paper, we show how rate guarantees to UBR can improve TCP performance in the presence of higher priority traffic.

- *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 design options exist for GFR, including network policing, per-VC scheduling and intelligent buffer management. Currently, very few implementations have been proposed for GFR and none have been tested for satellite latencies. 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. In this paper, we focus on TCP performance over UBR and GFR services.

In addition to the network-based options, there are four TCP congestion control techniques that are of interest in performance analysis over satellite links [2]:

- Slow start and congestion avoidance (TCP Vanilla).
- Fast retransmit and recovery (TCP Reno).
- TCP New Reno.
- Selective acknowledgments (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 [2]. However, these studies have focussed only on TCP mechanisms and have not considered intelligent network-based traffic management or guaranteed rate policies. Also, the studies have all been performed using a best effort service framework. Future broadband satellite networks must support the multiservice framework being adopted for terrestrial networks. Satellite networks using an ATM-like cell transport must use network-based techniques to provide the service guarantees required for a multiservice network.

3. PERFORMANCE ANALYSIS OF TCP OVER SATELLITE-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

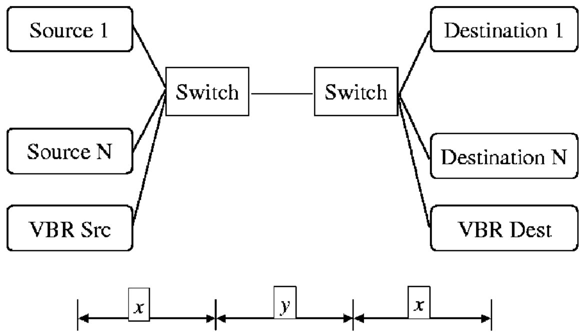

Figure 1. N-source TCP over satellite-UBR configuration with VBR background.

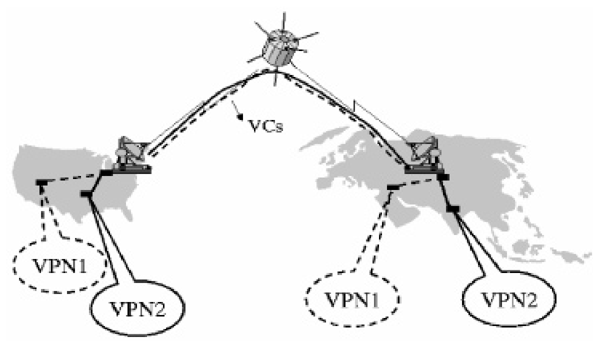

Figure 2. A GEO satellite used as a backbone network.

utilized efficiently to be cost effective. In this section, we are interested in analysing the effects of TCP flavors, buffer management policies and guaranteed rate on the performance TCP over UBR+ with satellite latencies, in the presence of higher priority variable bit rate (VBR) traffic sharing the link. We now describe the configuration used in our simulations, the performance metrics and the results of our experiments.

3.1. Simulation configuration

Figure 1 illustrates a sample network configuration used in this study. We use this model to simulate both GEO and LEO systems. The figures shows a general scenario with TCP sources running over ATM. The two ATM switches shown in the figure can be either on-board or ground-based. The total end-to-end delay is modeled as either a GEO or a LEO delay.

For the GEO configuration, the two switches represent earth stations connected by a GEO hop. The round trip latency between two earth stations is 550 ms. Figure 2 illustrates an example deployment scenario with the GEO satellite serving as the backbone connecting virtual private networks (VPNs).

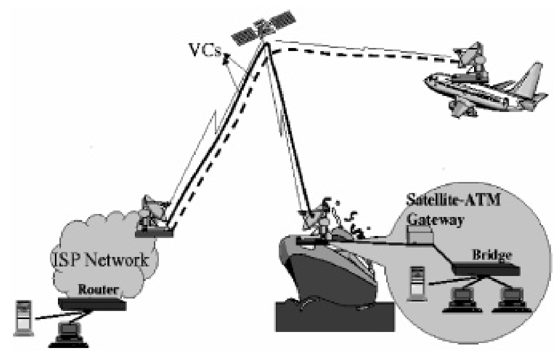

Figure 3. A LEO satellite providing remote access to nearby offshore networks.

For the LEO configuration, the switches may represent on-board switches or ground-based switches. The uplink and downlink delays are 5 ms each and correspond to satellites at about 700 km altitude and 60° elevation angle [15]. Figure 3 illustrates how the sample configuration corresponds to a LEO system providing remote access to offshore networks. The uplink delay, downlink delay and the delay through the terrestrial ISP are all 5 ms each. The first switch represents an on-board switch while the second switch represents a terrestrial switch. The total round trip latency is 30 ms. The simulations in Section 4 also use a multiple hop LEO system with a round trip latency of 120 ms. To highlight the effects of the factors for different delays, in Section 3.3, we also use a very short delay of 30 μs. The configuration reflects a campus or an office network with no satellite component.

Note that the LEO and GEO configurations should not be compared by themselves. The coverage provided by the LEO is much less than that provided by the GEO. The reason for selecting these configurations, is to highlight the effect of latency on TCP performance. The LEO configuration represents a lower bound on the satellite propagation delay, while the GEO configuration represents an upper bound.

In our simulation, the buffering delays and packet losses due to congestion occur only at a single multiplexing point in the network. In practice, queues may occur anywhere in the network, but our assumption simplifies the simulation without restricting the results. We also assume a constant delay without any jitter. The values chosen for uplink and downlink delays are consistent with typical constellations presented in Reference [15].[§]

The latency values are summarized below:

- *GEO*: Round trip latency = 550 ms.
- *LEO*: Round trip latency = 30 ms.
- *Multiple hop LEO*: Round trip latency = 120 ms (Section 4).
- *Negligible delay* (*LAN*): Round trip latency = 30 μs (Section 3.3).

[§] Note that TCP performance is dependent only on the total round trip latency regardless of the individual uplink and downlink delays. Also, the location of the queues does not affect the performance. In our LEO simulations, the queuing point is 5 ms away from the source earth terminals.

In the simulation results presented in Section 3.3, an additional variable bit rate (VBR) source is also present. All sources except the VBR source, are identical and infinite TCP sources. The TCP layer always sends 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 VBR source is also an end to end source like the other TCP connections. The VBR source is an on–off source with equal on and off periods of 300 ms each. During the on time, the source uses up the entire link capacity (155.52 Mbps). The on–off period was selected based on preliminary simulations presented in Reference [14]. Similar on–off patterns have also been used in Reference [16].

The TCP maximum segment size (MSS) is set to 512 bytes for the LAN and LEO configurations. This is the common segment size used in most current TCP implementations. For the GEO configuration, we use a segment size of 9180 bytes.¶ For the LAN configurations, the TCP maximum window size is limited by a receiver window of 64K bytes. This is the default value specified for TCP implementations. For LEO configurations, a window of 64K bytes is not sufficient to achieve 100 per cent utilization. We thus use the window scaling option to specify a maximum window size of 600 000 bytes. For GEO configurations, this value is further scaled up to 8704 000 bytes.

All link bandwidths are 155.52 Mbps. The Duration of the simulation is 10 s for LANs, 20 s for LEOs and 40 s for GEOs. This allows for adequate round trips for the simulation to give stable results.

The ATM switch (either on-board or ground-based) is output buffered, where each output port has a separate buffer for each service category (or class). Figure 4 shows the switch model. The switch supports multiple service categories as shown in the figure. Each service category is provided with a bandwidth guarantee. In this section, we consider only two classes—VBR and UBR. VBR typically has strict priority over UBR, but UBR may be guaranteed a fraction of the total link capacity. This fraction is called the guaranteed rate (GR). In Section 5 we consider the GFR service that has its own queue and is guaranteed a minimum rate. This is called the *GFR capacity*. The GFR capacity is allocated among the GFR VCs as their MCRs.

To enforce a GR (as a fraction of the total link capacity), we perform fair scheduling among the queues on each port. The fair scheduling algorithm ensures that when $\mathrm{GR} > 0.0$, the UBR queue is never starved, i.e. on the average, for every N cells transmitted on to the link, $\mathrm{GR} \times N$ cells are from the UBR queue. This means that the VBR cells could be queued if the VBR connections are using more than $(1 - \mathrm{GR})$ of the link capacity. Any capacity unused by VBR is also allocated to UBR. In our simulations, we use three values of GR—0, 10 and 50 per cent.

The simulations are performed for two values of the number of sources (N) and buffer sizes in the switches as described in Section 3.3.

3.2. Performance metrics

When ATM networks carry TCP 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

¶ We use a large value of MSS for GEO to avoid the division by zero problem in implementations of the congestion avoidance algorithm in TCP. This problem only occurs for large values of congestion windows. More information is given in Reference [17].

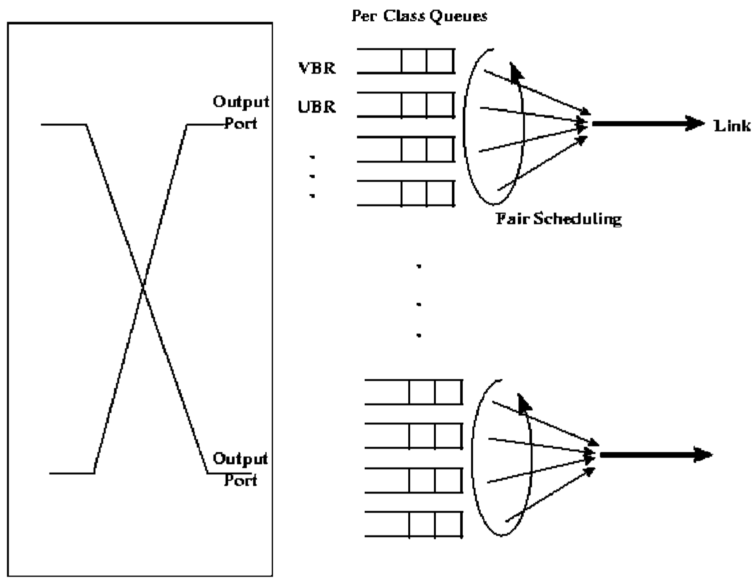

Figure 4. Switch model for UBR + with GR. The model assumes an output buffered switch with a single queue for each class. A fair scheduling mechanism provides a guaranteed rate to the UBR class.

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/s. 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.

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 Reference [18].

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 TCP MSS. For 9180 bytes of data, the ATM layer receives 9180 bytes of data with 20 bytes of TCP header, 20 bytes of IP header, 8 bytes of LLC header and 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 $\approx 9180/10\,229 \approx 89.7$ per cent ≈ 135 Mbps. 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. We measure fairness for TCP over a best effort service 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

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

where x_i is the observed throughput of the ith TCP connection ($0 < i \leqslant N$), and e_i the expected throughput or fair share for the ith TCP connection.

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 $x_1 = x_2 = \cdots = x_n$ 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 Reference [19]. This fairness index has been used in several studies including Reference [18]. 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 Reference [12], 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.

3.3. Analysis technique

In this section, we describe our analysis technique of the effects of the following four factors on TCP performance over UBR + :

- *Factor A. drop policy*: Tail Drop, EPD and SD.
- *Factor B. TCP type*: Vanilla, Reno and SACK
- *Factor C. Buffer size*: Thousand and 3000 cells for LANs, 12 000 and 36 000 cells for LEOs; and 200 000 and 600 000 cells for GEOs.
- *Factor D. Guaranteed rate* zero, 10 and 50 per cent of the link capacity.

The effect of the factors is analysed for three values of the round trip time—30 μs (LAN), 30 ms (LEO) and 550 ms (GEO).

We performed 2 sets of experiments with 5 and 15 TCP sources. In addition to the TCP sources, an on-off VBR source with a period of 300 ms was also present in the system.

We performed a full factorial experiment with the above factors and analysed the effect of each factor on the efficiency of the system. Tables II–IV in the Appendix list the complete results of the simulations.

To quantify the effect of each factor in our experiment, we use the allocation of variation technique described in Reference [19]. Only a brief description of the methodology is presented here.

The goal of analysing results of the experiments was to calculate the individual effects of contributing factors and the interactions between the factors. These effects can also help us in drawing meaningful conclusions about the optimum values for different factors. In our experiments, we analyse the effects of the TCP flavours, buffer sizes, drop policies and guaranteed rate in determining the efficiency and fairness for LAN, LEO, and GEO links. In this experiment, there were 4 factors—drop policy (A), TCP flavour (B), switch buffer (C) and guaranteed rate (D). The values a factor can take are called *levels* of the factor. For example, EPD and SD are two levels of the factor *drop policy*. For factors A, B, C and D, the levels are indicated by i, j, k and l, respectively. Each simulation corresponds to a combination of the levels of the factors. *A full factorial* experiment calculates performance results (in our case efficiency and fairness) for all such combinations. In this case, for each set of sources (5 and 15), the total number of experiments is 162. We performed 54 experiments for each configuration (LAN, LEO and GEO) for a total of 162 simulations. The analysis was done separately for LAN, LEO and GEO configurations.

Table I. Allocation of variation: efficiency.*

	LAN		LEO		GEO	
Component	Value	% of SST	Value	% of SST	Value	% of SST
SST	2.51	100	1.27	100	3.31	100
SSY	29.00		38.57		29.50	
SSO	26.48		37.30		26.19	
SSA	1.12	$\boxed{44.8}$	0.03	2.77	0.15	4.67
SSB	0.21	8.44	0.02	2.03	2.45	$\boxed{74.17}$
SSC	0.58	$\boxed{23.10}$	0.37	$\boxed{29.21}$	0.09	2.78
SSD	0.06	2.60	0.52	$\boxed{41.28}$	0.09	2.74
SSAB	0.05	2.10	0.01	1.14	0.21	6.58
SSAC	0.05	1.99	0.008	0.6	0.002	0.06
SSAD	0.04	1.91	0.01	1.12	0.02	0.76
SSBC	0.08	3.30	0.03	2.80	0.02	0.81
SSBD	0.02	0.82	0.09	7.20	0.09	2.89
SSCD	0.04	1.84	0.09	7.44	0.02	0.83
SSE	0.22	9.00	0.05	4.00	0.12	5.00

* The numbers in the *Value* column are calculated using the equations in Section 3.3. In LANs, drop policy and buffer size (A and C) explain most of the variation. In LEOs, guaranteed rate (D) is the most important factor, while buffer size also explains some variation. For GEO satellites, TCP flavour (B) is the most important factor in the allocation of variation.

We assume an additive model given by the following equation:

$$y = \mu + \alpha_i + \beta_j + \zeta_k + \delta_l + \gamma_{ij} + \gamma_{ik} + \gamma_{il} + \gamma_{jk} + \gamma_{jl} + \gamma_{kl} + \gamma_{ijk} + \gamma_{ikl} + \gamma_{jkl} + \gamma_{ijkl} + \varepsilon_{ijkl}$$

where i, j, k and l are the levels of factors A, B, C and D, respectively.

The model (y) consists of the sum of the mean response (μ), 4 main effects (α_i, β_j, ζ_k and δ_l), 6 first-order interactions (γ_{ij}, γ_{ik}, γ_{il}, γ_{jk}, γ_{jl} and γ_{kl}), 3 second-order interactions (γ_{ijk}, γ_{ikl}, γ_{jkl} and γ_{ijkl}), 1 third-order interaction (γ_{ijkl}) and an experimental error term (ε_{ijkl}). We assume that only first-order interactions are significant; second- and third-order interactions are ignored.

We calculate the following quantities (see Table I):

Observation or response (y_{ijkl}): This is the efficiency or fairness from an experiment with the levels of individual factors as i, j, k and l, respectively.

Sum of squares of responses (SSY): This is the sum of squares of the individual results above.

Sum of squares of the overall mean (SSO): This consists of the calculation of the overall mean, $\bar{y}$, of the results y_{ijkl} and multiplying its square by the total number of experiments.

Total variation (SST): This represents the variation in the result values (efficiency or fairness) around the overall mean

$$\text{SST} = \text{SSY} - \text{SSO}$$

Sum of squares of main effects (SSA SSB, SSC, SSD): The main effects (α_i, β_j, ζ_k and δ_l) are the individual contributions of a level of each factor (A, B, C and D) to the overall result. A particular main effect is associated with a level of a factor and indicates how much variation around the overall mean is caused by the level:

$$\alpha_i = \bar{y}_{.jkl} - \mu$$

$$\beta_j = \bar{y}_{i.kl} - \mu$$

$$\zeta_k = \bar{y}_{ij.l} - \mu$$

$$\delta_l = \bar{y}_{ijk.} - \mu$$

$$\text{SSA} = bcd \times \sum_i \alpha_i^2$$

$$\text{SSB} = acd \times \sum_j \beta_j^2$$

$$\text{SSC} = abd \times \sum_k \zeta_k^2$$

$$\text{SSD} = abc \times \sum_l \delta_l^2$$

where a, b, c and d are the number of levels of factors A, B, C and D, respectively.

First-order interactions (SSAB, SSAC, etc.): These are the interactions between levels of two factors. In the example, there are first order interactions between each TCP flavour and buffer size, between each drop policy and TCP flavour, between each buffer size and drop policy, TCP flavour and guaranteed rate and so on. For example, the first-order interaction term between drop policy (A) and TCP flavour (B) is given by

$$\gamma_{ij} = \bar{y}_{ij..} - \bar{y}_{i...} - \bar{y}_{.j..} - \mu$$

$$\mathrm{SSAB} = cd \times \sum_{i,j} (\gamma_{ij})^2$$

Sum of squares of overall standard error (SSE): This represents the experimental error associated with each result value. The overall standard error is also used in the calculation of the confidence intervals for each effect:

$$\mathrm{SSE} = \mathrm{SSY} - \mathrm{SSO} - \mathrm{SSA} - \mathrm{SSB} - \mathrm{SSC} - \mathrm{SSD} - \mathrm{SSAB} - \mathrm{SSAC}$$
$$- \mathrm{SSAD} - \mathrm{SSBC} - \mathrm{SSBD} - \mathrm{SSCD}$$

Allocation of variation: This is used to explain how much each effect contributes to the total variation (SST):

$$\mathrm{SST} = \mathrm{SSA} + \mathrm{SSB} + \mathrm{SSC} + \mathrm{SSD} + \mathrm{SSAB} + \mathrm{SSAC}$$
$$+ \mathrm{SSAD} + \mathrm{SSBC} + \mathrm{SSBD} + \mathrm{SSCD} + \mathrm{SSE}$$

Each term on the right of the above equation contributes to the total variation. An effect (a factor or interaction), which explains a large fraction of the total variation, is said to be important.

Confidence intervals for main effects: The 90 per cent confidence intervals for each main effect are calculated. If a confidence interval contains 0, then the corresponding level of the factor is not statistically significant. If confidence intervals of two levels overlap, then the effects of both levels are assumed to be similar.

3.4. Results and discussion

Table I shows the results of the allocation of variation for efficiency. The results show that the model is applicable to efficiency because most of the variation is explained by the main effects and the interactions.

For LAN, the most important factors are the drop policy that explains 44 per cent of the variation and the buffer size that explains 23 per cent of the variation. The results show that large buffer sizes with selective drop produce the best efficiency. For LEO, the most important factors are the buffer size (41 per cent of variation) and the TCP type (29 per cent of variation). Large buffer and SACK produce the best performance. For GEO, TCP type is the most important factor (explains 74 per cent of the variation). SACK provides the best performance. The interactions between the factors are insignificant.

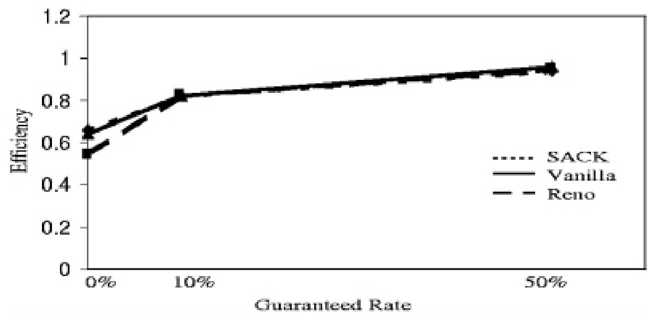

Figure 5. LEO: guaranteed rate versus TCP.

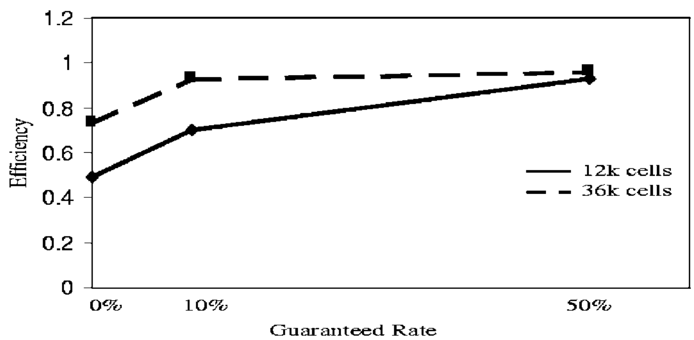

Figure 6. LEO: guaranteed rate versus buffer size.

Figure 5 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. Even a 10 per cent bandwidth reservation can increase the overall throughput by about 25 per cent.

Figure 6 shows the relative effects of GR and buffer size on LEO efficiency. Each point in the figure represents the efficiency value averaged over all the other factors (number of sources, drop policy and TCP mechanism). A 10 per cent GR allocation increases the efficiency by about 20 per cent. A larger buffer size (36K cells) along with 10 per cent GR can provide high efficiency.

Figure 7 illustrates the corresponding result for GEO delays. The effect of GR is insignificant relative to the effect of TCP. Reno performs very poorly, while SACK performs the best.

From the analysis, the following results can be summarized for TCP over UBR+ with GR in the presence of high priority background traffic.

Result 1 (LAN (30 μs))

For LANs, the dominating factors that affect the performance are the switch drop policy and the buffer size.

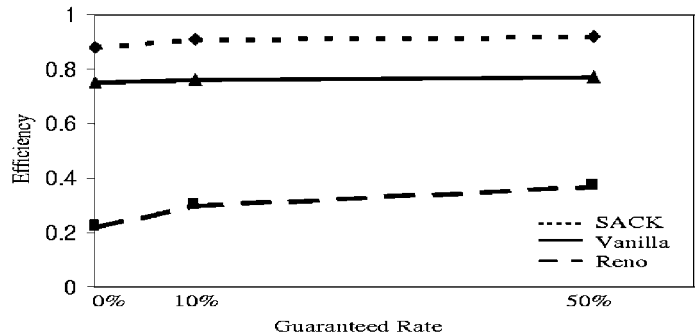

Figure 7. GEO: guaranteed rate versus TCP.

The selective drop policy improves the performance irrespective of most TCP and GR parameters. This result holds with or without the presence of background VBR traffic. In LANs, the switch buffer sizes are of the order of 1000 and 3000 cells. This is very small in comparison with the maximum TCP receiver window (64K bytes). As a result, TCP can easily overload the switch buffers. This makes buffer management very important for LANs.

Result 2 (LEO (30 ms))

For LEOs, the dominating factor is the GR.

GR values of 0.5 and 0.1 produce the highest efficiency values. A constant amount of bandwidth provided by GR ensures that TCP keeps receiving ACKs from the destination. This reduces the variability in the round trip times. Consequently, TCP is less likely to timeout. Buffer management policies do have an impact on TCP performance over LEOs, but the effect is less than in LANs. This is because the buffer sizes of LEO switches are comparable to the bandwidth $\times$ round trip delays of the network. The TCP maximum windows are also usually based on the round trip times. As a result, buffers are more easily available and drop policies are less important.

Result 3 (GEO (550 ms))

For GEO networks, the TCP congestion control mechanism makes the most difference — SACK TCP produces the best results and Reno TCP results in the worst performance.

SACK TCP ensures quick recovery from multiple packet losses, whereas fast retransmit and recovery is unable to recover from multiple packet drops. The switch buffer sizes are quite large and so the drop policies do not make a significant difference. The GR fractions do not significantly affect the TCP performance because in our simulations, the VBR burst durations are smaller than the round trip propagation delays. The retransmission timeout values are typically close to 1 s and so a variation of the RTT by 300 m can be tolerated by the TCP. GR may have more impact on GEO networks in cases where UBR is starved for times larger than the retransmission timeout value of the connection. However, a VBR on-off period of more than 1 s is not a realistic model.

4. BUFFER REQUIREMENTS FOR TCP OVER SATELLITE-UBR +

Previous studies have shown that small switch buffer sizes can result low TCP throughput over UBR [6]. 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 analysed the effect of buffer sizes on performance. *As a result, it is not clear how the increase in buffers affects throughput and what buffer sizes provide the best cost–performance benefits for TCP over UBR +*. In this section, we present simulation experiments to assess the buffer requirements for three satellite delay–bandwidth products for TCP over UBR +.

4.1. Simulation parameters

In this experiment, we use the N-source configuration without the VBR source. We study the effects of the following factors:

- *Round trip latency*: We use three round trip values: 550 ms (GEO), 120 ms (multiple hop LEO), and 30 ms (single LEO).
- *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), $k = -1, \ldots, 6$ (i.e. 2, 1, 0.5, 0.25, 0.125, 0.0625, 0.031 and 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, 6K, 12K (= 1 RTT), 24K and 36K.
  — *Multiple LEO (120 ms)*: 780, 1560, 3125, 6250, 12.5K, 50K (= 1 RTT) and 100K.
  — *GEO (550 ms)*: 3375, 6750, 12 500, 25K, 50K, 100K, 200K (= 1 RTT) and 400K.

- *Switch drop policy and TCP policy*: To restrict the number of factors, we analyse the best case scenario by using selective drop and TCP SACK as the drop policy and the TCP policy, respectively.

All other parameters are the same as before. We plot the buffer size against the achieved TCP throughput for different delay–bandwidth products and numbers of sources. The asymptotic nature of this graph provides information about the optimal buffer size for the best cost–performance ratio.

4.2. Results and discussion

Figures 8–10 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, ($0.016 \times$ RTT, $0.031 \times$ RTT, $0.0625 \times$ 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 than 1 round trip delay–bandwidth), TCP

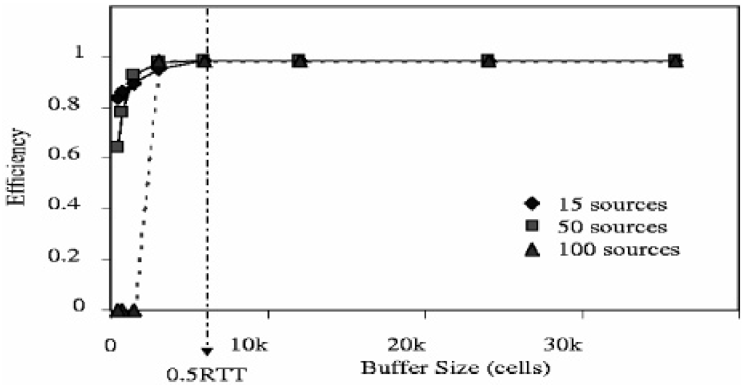

Figure 8. Buffer requirements for single hop LEO.

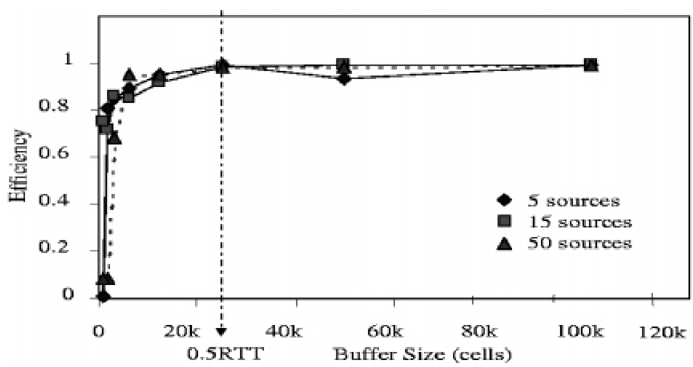

Figure 9. Buffer requirements for multiple hop LEO.

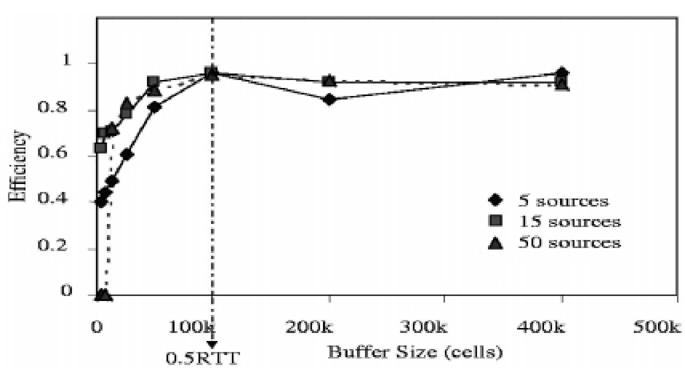

Figure 10. Buffer requirements for GEO.

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 per cent, but for buffers greater than $0.5 \times$ RTT, the average TCP throughput is over 98 per cent 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.

The optimal buffer size from the graphs is about one-half of the round trip delay–bandwidth product of the network. Intuitively, this result can be explained as follows. The network capacity is determined by the product of the bottleneck bandwidth and the round trip delay. During steady state, all the TCPs share this bottleneck capacity by adjusting their window sizes, such that the sum of the TCP windows equals the round trip bandwidth–delay product of the network. Thus, at any given time, the total sequence number space of outstanding TCP segments is limited by this number. If we assume a steady flow of packets and ACKs, then at any given time, one-half of the outstanding sequence space is in the form of packets travelling in the forward direction, while the other half is in the form of ACKs travelling in the opposite direction. If all the segments arrive at the bottleneck at about the same time, the queue at the bottleneck must be able to buffer all these packets. Thus, $0.5\text{RTT} \times$ bandwidth product worth of buffer provides good TCP performance. In the worst case, the TCP traffic could be highly bursty and the ACKs could be out of phase with the segments. All sources would send the entire window of packets and wait for all the ACKs to arrive before sending another window. In this case, the buffer requirements would be equal to the sum of all the TCP maximum window sizes.

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 9 and 10). In our simulations, we see more timeouts with these buffer sizes. 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. The effect is more pronounced for a small number a sources because after a timeout, their windows must reach a large value to achieve full network capacity. With more connections, smaller individual windows are enough to fill the pipe. The extra time it takes for the windows to increase to full capacity results in the loss of throughput.

Result 4 (Buffer requirements)

The simulations show that a buffer size of 0.5RTT at the bottleneck provides high efficiency and fairness to TCPs over UBR+ for satellite networks.

5. BUFFER MANAGEMENT FOR GUARANTEED FRAME RATE OVER SATELLITE

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 if a frame is eligible for MCR guarantees. Such frame are called QoS eligible. The minimum rate is guaranteed to the $\mathrm{CLP} = 0$ 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 Reference [20], but the above discussion captures the essence of the service.

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 [20]. Several proposals have been made [21,22] 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 queuing and scheduling were 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 Reference [23].

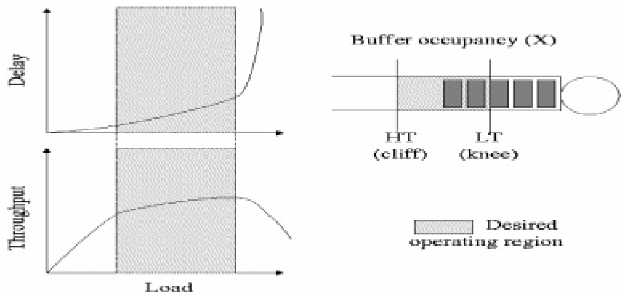

Figure 11. DFBA target operating region.

5.1. *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 11 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.

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 $\mathrm{VC}_i = \mathrm{MCR}_i/(\text{GFR capacity})$,

$W = \Sigma W_i$,

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

Z_i = parameter ($0 \leqslant Z_i \leqslant 1$).

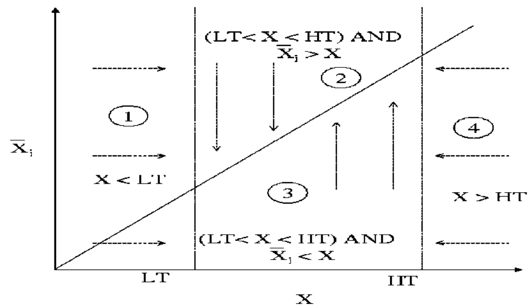

Figure 12. DFBA drop regions.

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 12 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.

In region 2, the packets of VC_i are dropped in a probabilistic manner. This drop behaviour is controlled by the drop probability function $P\{\text{drop}\}$. 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 [24] and FRED [25]. 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 behaviour. 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 Reference [23] and is given by

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

For satellite latencies, an important parameter in this equation is Z_i. It has been shown [26] 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 [26], 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

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

The date 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 date 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 satellite 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 total delay. Terrestrial switches are limited in this respect. This limitation is also discussed in Reference [18].

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 VC. 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.

5.2. Simulation results

The test results presented here are with DFBA for Satellite-ATM interconnected TCP networks. Figure 13 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. 353 207 cells per s). The MCRs were 20, 40, 60, 80 and 100 kcells/s for VCs 1–5, respectively, giving a total MCR allocation of 85 per cent 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 and 34.56 Mbps, respectively.

‖ The figure only shows two pairs of local switches, but our simulations had 5.

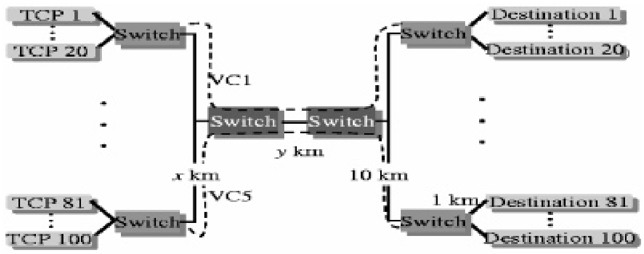

Figure 13. DFBA simulation configuration.

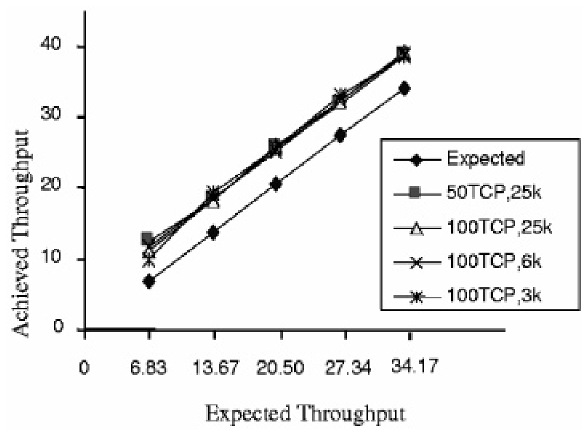

Figure 14. Performance of DFBA for terrestrial networks.

Note that, in GFR deployments, MCRs are expected to be allocated more conservatively and 85 per cent allocation reflects an upper bound on MCR allocation. Also, these numbers are aggregate numbers for all 20 TCPs for VCs 1–5. All TCP sources are persistent TCPs with SACK. Based on previous studies, 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 Reference [23].

In Figure 13, 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 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 – 25 000, 6000 and 3000 cells. The goal of this experiment is to illustrate that DFBA achieves the MCR guarantees for each VC. Figure 14 illustrates the expected and achieved throughputs for each VC in the configuration. The achieved throughput for a VC is the sum of all the TCP throughputs in that VC. The figure 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 per cent resulting in high network

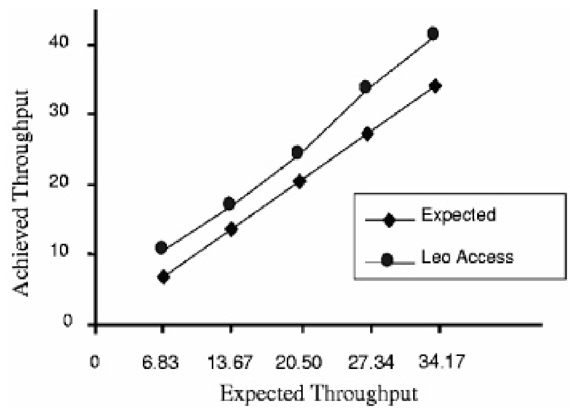

Figure 15. Performance of DFBA for LEO access.

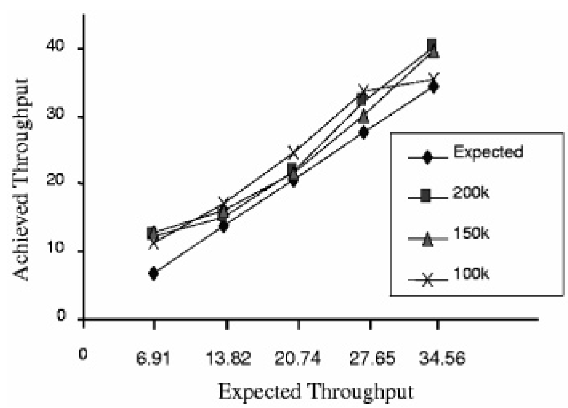

Figure 16. Performance of DFBA for GEO backbone.

utilization. In the simulations, the excess capacity (GFR capacity — MCR allocation) is almost equally distributed among the five VCs.

- *LEO Access with heterogenous RTT*: In this configuration, the access hop (x) for VC 3, is a LEO hop 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 their backbone delays are also 5 ms one way. The results of this simulation with buffer size = 6000 cells is shown in Figure 15. 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 about 550 ms. The GEO hop is the most dominant hop with respect to latency and the access hops had negligible latency. Figure 16 shows the achieved throughputs for three different buffer sizes — 200 000, 150 000 and 100 000 cells: 100 000 cells corresponds to 0.5RTT-bandwidth product worth of buffers. Again, the figure shows that DFBA provides MCR guarantees to VCs over long delay satellite networks.

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

Results 5 (GFR service)
The Guaranteed frame rate service is designed for frame-based best effort applications and supports per-VC minimum cell rate guarantees.

Result 6 (GFR 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.

Result 7 (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.

Result 8 (Limitations)
In general, buffer management schemes for TCP are limited by the granularity of IP or ATM flows.

6. SUMMARY OF RESULTS

This paper describes a set of techniques for improving the performance of TCP 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 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 addressed in this paper are:

- UBR with frame level discard policies,
- UBR with intelligent buffer management,
- UBR with guaranteed rate.

The following TCP mechanisms were studied:

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

We also used the guaranteed frame rate (GFR) to provide minimum cell rate guarantees to VCs carrying TCP traffic. We proposed the differential fair buffer allocation (DFBA) algorithm for buffer management of TCP over GFR.

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 losses 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-VC accounting is used.
4. Throughput increases further with more aggressive 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. 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 10 per cent, of the bandwidth for UBR significantly improves TCP performance. For GEO systems, the effect of TCP SACK was more significant than other factors.
6. A buffer size equal to about half the round-trip delay-bandwidth product of the TCP connections provides high efficiency for TCP over satellite-UBR.
7. The GFR service category can provide per-VC MCR guarantees. The proposed differential fair buffer allocation (DFBA) scheme used per-VC accounting to provide MCR guarantees to VCs carrying TCP traffic.

The results described above have been based on simulations using persistent TCP traffic. In Reference [11], we have studied the performance of TCP over satellite-UBR+ using a WWW model for TCP traffic. The results obtained from the study are consistent with those presented in this paper.

ACKNOWLEDGEMENTS

This work was partially sponsored by the NASA Lewis Research Center under contract number NAS3-97198.

APPENDIX

The complete results of simulations are given in Tables II–VI.

Table II. TCP over UBR+ with VBR (300 ms on/off): LAN.

Number of sources	Buffer (cells)	TCP	GR	UBR	EPD	Selective drop
5	1000	SACK	0.5	0.26	0.85	0.96
5	1000	SACK	0.1	0.98	0.57	0.75
5	1000	SACK	0.0	0.71	0.88	0.98
5	3000	SACK	0.5	0.96	0.97	0.95
5	3000	SACK	0.1	0.93	0.89	0.99
5	3000	SACK	0.0	0.83	0.91	0.92
5	1000	Reno	0.5	0.22	0.30	0.61
5	1000	Reno	0.1	0.37	0.41	0.66
5	1000	Reno	0.0	0.14	0.92	0.39

Table II. *Continued.*

Number of sources	Buffer (cells)	TCP	GR	UBR	EPD	Selective drop
5	3000	Reno	0.5	0.60	0.69	0.76
5	3000	Reno	0.1	0.55	0.79	0.93
5	3000	Reno	0.0	0.59	0.72	0.92
5	1000	Vanilla	0.5	0.46	0.47	0.58
5	1000	Vanilla	0.1	0.40	0.58	0.70
5	1000	Vanilla	0.0	0.27	0.73	0.80
5	3000	Vanilla	0.5	0.88	0.72	0.87
5	3000	Vanilla	0.1	0.61	0.63	0.90
5	3000	Vanilla	0.0	0.61	0.88	0.85
15	1000	SACK	0.5	0.38	0.74	0.92
15	1000	SACK	0.1	0.49	0.76	0.91
15	1000	SACK	0.0	0.57	0.98	0.90
15	3000	SACK	0.5	0.90	0.96	0.92
15	3000	SACK	0.1	0.61	0.94	0.96
15	3000	SACK	0.0	0.43	0.86	0.95
15	1000	Reno	0.5	0.43	0.52	0.70
15	1000	Reno	0.1	0.35	0.48	0.68
15	1000	Reno	0.0	0.29	0.40	0.70
15	3000	Reno	0.5	0.68	0.88	0.95
15	3000	Reno	0.1	0.63	0.81	0.97
15	3000	Reno	0.0	0.54	0.69	0.89
15	1000	Vanilla	0.5	0.59	0.42	0.80
15	1000	Vanilla	0.1	0.38	0.52	0.70
15	1000	Vanilla	0.0	0.36	0.39	0.75
15	3000	Vanilla	0.5	0.68	0.90	0.97
15	3000	Vanilla	0.1	0.54	0.96	0.98
15	3000	Vanilla	0.0	0.37	0.85	0.89

Table III. TCP over UBR + with VBR (300 ms on/off): LEO.

Number of sources	Buffer (cells)	TCP	GR	UBR	EPD	Selective drop
5	12 000	SACK	0.5	0.95	0.93	0.94
5	12 000	SACK	0.1	0.87	0.66	0.69
5	12 000	SACK	0.0	0.42	0.43	0.61
5	36 000	SACK	0.5	0.97	0.99	0.99
5	36 000	SACK	0.1	0.96	0.98	0.96
5	36 000	SACK	0.0	0.55	0.52	0.96
5	12 000	Reno	0.5	0.93	0.96	0.94
5	12 000	Reno	0.1	0.61	0.79	0.71
5	12 000	Reno	0.0	0.34	0.45	0.33
5	36 000	Reno	0.5	0.97	0.97	0.93
5	36 000	Reno	0.1	0.90	0.96	0.75
5	36 000	Reno	0.0	0.33	0.92	0.33
5	12 000	Vanilla	0.5	0.94	0.97	0.96
5	12 000	Vanilla	0.1	0.82	0.70	0.69

Table III. *Continued.*

Number of sources	Buffer (cells)	TCP	GR	UBR	EPD	Selective drop
5	12 000	Vanilla	0.0	0.49	0.36	0.42
5	36 000	Vanilla	0.5	0.97	0.97	0.97
5	36 000	Vanilla	0.1	0.96	0.90	0.94
5	36 000	Vanilla	0.0	0.92	0.33	0.92
15	12 000	SACK	0.5	0.88	0.85	0.90
15	12 000	SACK	0.1	0.72	0.61	0.76
15	12 000	SACK	0.0	0.64	0.48	0.58
15	36 000	SACK	0.5	0.96	0.95	0.97
15	36 000	SACK	0.1	0.95	0.94	0.97
15	36 000	SACK	0.0	0.93	0.72	0.95
15	12 000	Reno	0.5	0.97	0.94	0.97
15	12 000	Reno	0.1	0.84	0.66	0.79
15	12 000	Reno	0.0	0.67	0.53	0.51
15	36 000	Reno	0.5	0.97	0.97	0.98
15	36 000	Reno	0.1	0.96	0.96	0.97
15	36 000	Reno	0.0	0.67	0.66	0.59
15	12 000	Vanilla	0.5	0.90	0.92	0.96
15	12 000	Vanilla	0.1	0.77	0.66	0.74
15	12 000	Vanilla	0.0	0.67	0.61	0.67
15	36 000	Vanilla	0.5	0.98	0.97	0.97
15	36 000	Vanilla	0.1	0.96	0.96	0.97
15	36 000	Vanilla	0.0	0.94	0.93	0.93

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

Drop policy	TCP	Buffer	GR	Efficiency	Fairness
Selective drop	SACK	200 000	0.5	0.87	0.91
Selective drop	SACK	200 000	0.1	0.78	0.82
Selective drop	SACK	200 000	0.0	0.74	0.87
Selective drop	SACK	600 000	0.5	0.99	1.00
Selective drop	SACK	600 000	0.1	0.99	0.99
Selective drop	SACK	600 000	0.0	0.99	1.00
Selective drop	Reno	200 000	0.5	0.33	0.71
Selective drop	Reno	200 000	0.1	0.24	0.93
Selective drop	Reno	200 000	0.0	0.16	1.00
Selective drop	Reno	600 000	0.5	0.35	0.99
Selective drop	Reno	600 000	0.1	0.39	0.99
Selective drop	Reno	600 000	0.0	0.30	0.98
Selective drop	Vanilla	200 000	0.5	0.83	0.90
Selective drop	Vanilla	200 000	0.1	0.71	0.99
Selective drop	Vanilla	200 000	0.0	0.81	0.87
Selective drop	Vanilla	600 000	0.5	0.79	1.00
Selective drop	Vanilla	600 000	0.1	0.80	0.99
Selective drop	Vanilla	600 000	0.0	0.76	1.00

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

Drop policy	TCP	Buffer	GR	Efficiency	Fairness
Early packet discard	SACK	200 000	0.5	0.84	1.00
Early packet discard	SACK	200 000	0.1	0.88	0.87
Early packet discard	SACK	200 000	0.0	0.82	0.99
Early packet discard	SACK	600 000	0.5	0.99	0.95
Early packet discard	SACK	600 000	0.1	0.99	0.88
Early packet discard	SACK	600 000	0.0	0.99	1.00
Early packet discard	Reno	200 000	0.5	0.46	0.51
Early packet discard	Reno	200 000	0.1	0.26	0.89
Early packet discard	Reno	200 000	0.0	0.17	0.99
Early packet discard	Reno	600 000	0.5	0.36	0.96
Early packet discard	Reno	600 000	0.1	0.34	0.98
Early packet discard	Reno	600 000	0.0	0.28	0.98
Early packet discard	Vanilla	200 000	0.5	0.71	1.00
Early packet discard	Vanilla	200 000	0.1	0.76	0.85
Early packet discard	Vanilla	200 000	0.0	0.68	1.00
Early packet discard	Vanilla	600 000	0.5	0.78	0.99
Early packet discard	Vanilla	600 000	0.1	0.80	0.99
Early packet discard	Vanilla	600 000	0.0	0.77	0.98

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

Drop policy	TCP	Buffer	GR	Efficiency	Fairness
UBR	SACK	200 000	0.5	0.87	0.91
UBR	SACK	200 000	0.1	0.87	1.00
UBR	SACK	200 000	0.0	0.85	1.00
UBR	SACK	600 000	0.5	0.93	0.85
UBR	SACK	600 000	0.1	0.96	0.87
UBR	SACK	600 000	0.0	0.90	0.96
UBR	Reno	200 000	0.5	0.87	0.88
UBR	Reno	200 000	0.1	0.36	0.92
UBR	Reno	200 000	0.0	0.38	0.9
UBR	Reno	600 000	0.5	0.84	0.84
UBR	Reno	600 000	0.1	0.69	0.77
UBR	Reno	600 000	0.0	0.47	0.98
UBR	Vanilla	200 000	0.5	0.87	0.84
UBR	Vanilla	200 000	0.1	0.73	1.00
UBR	Vanilla	200 000	0.0	0.84	0.86
UBR	Vanilla	600 000	0.5	0.83	0.99
UBR	Vanilla	600 000	0.1	0.83	0.99
UBR	Vanilla	600 000	0.0	0.81	1.00

REFERENCES

1. IETF TCP over Satellite working groups. http://tcpsat.lerc.nasa.gov
2. Allman M. (ed). *Ongoing TCP Research Related to Satellites*, *Internet Draft*, work in progress.
3. Ponzoni C. High data rate on-board switching. *Third Ka-band Utilization Conference*, September 1997.

4. Goyal R, Jain R, Kalyanaraman S, Fahmy S, Vandalore B. Improving the performance of TCP over the ATM-UBR service. *Computer Communications* 1998; **21**(10):898–911, http://www.cis.ohio-state.edu/jain/papers/cc.htm
5. Heinanen J. A fair buffer allocation scheme. Unpublished manuscript.
6. Romanov A, Floyd S. Dynamics of TCP traffic over ATM networks. *IEEE Journal of Selected Areas in Telecommunications*, 1995.
7. Li H, Siu KY, Tzeng HT, Ikeda C, Suzuki H. TCP over ABR and UBR services in ATM. *Proceeding IPCCC'96*, March 1996.
8. Jacobson V. Congestion avoidance and control. *Proceedings of the SIGCOMM'88 Symposium*, August 1988, 314–332.
9. Stevens W. TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms. *Internet RFC* 2001, January 1997.
10. Mathis M, Madhavi J, Floyd S, Romanow A. TCP selective acknowledgment options. *Internet RFC* 2018, October 1996.
11. Goyal M, Goyal R, Jain R, Vandalore B, Fahmy S, VonDeak T, Bhasin K, Butts N, Kota S. 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.
12. Li H, Siu KY, Tzeng HT, Ikeda C, Suzuki H. TCP over ABR and UBR services in ATM. *Proceedings IPCCC'96*, March 1996.
13. Sally Floyd's homepage http://www-nrg.ee.lbl.gov/floyd/epd.html
14. Goyal R, Jain R, Kalyanaraman S, Fahmy S, Vandalore B, Cai X, Kim S-C, Kota S. Guaranteed rate for improving TCP performance on UBR + over terrestrial and satellite networks. *ATM Forum*/97-0424, April 1997, http://www.cis.ohio-state.edu/jain/atmf/a97-044.htm
15. Satellite altitudes taken from Lloyd's satellite constellation. http://www.ee.surrey.ac.uk/Personal/L.Wood/constellations/overview.html
16. Kalyanaraman S. Traffic management for the available bit rate (ABR) service in asynchronous transfer mode (ATM) networks. *Ph.D. Dissertation*, The Ohio State University, 1997; xxiv + 429.
17. Goyal R, Jain R, Kalyanaraman S, Fahmy, S, Vandalore B, Kota S. Selective acknowledgments and UBR + drop policies to improve TCP/UBR performance over terrestrial and satellite networks. *Proceedings IC3N'97*, September 1997, (All our papers and ATM Forum contributions are available from http://www.cis.ohio-state.edu/~jain).
18. Floyd S. Connections with multiple congested gateways in packet-switched networks Part 1: one-way traffic. *Computer Communication Review* 1991; **21**(5):30–47.
19. Jain R. *The Art of Computer Systems Performance Analysis*: *Techniques for Experimental Design*, *Simulation and Modeling*. Wiley-Interscience: New York, April 1991.
20. *ATM Forum*. ATM Traffic Management Specification Version 4.1, December 1998.
21. Basak D, Pappu S. GFR implementation alternatives with fair buffer allocation schemes. *ATM Forum 97-0528*.
22. Bonaventure O. A simulation study of TCP with the proposed GFR service category. *DAGSTUHL Seminar 9725,* High Performance Networks for Multimedia Applications. Germany, June 1997.
23. Goyal R, Jain R, Fahmy S, Vandalore B. Buffer management for the GFR service. *ATM Forum*/98-0405 July 1998, http://www.cis.ohio-state.ed/jain/atmf/a98-0405.htm
24. Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE*/*ACM Transactions on Networking* 1993.
25. Lin D, Morris R. Dynamics of random early detection. *Proceedings of SIGCOMM97*, 1997.
26. Mathis M, Se?mke J, Mahdavi J, Ott T. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review* 1997; **27**(3):67–82.
27. Kota S, Goyal, R, Jain R. Satellite ATM network architectural considerations and TCP/IP performance. *Proceedings of the Third Ka Band Utilization Conference*, Italy, September 15–18, 1997; 481–488, http://www.cis.ohio-state.edu/jain/papers/kaband.htm

