

\*\*\*\*\*

ATM Forum Document Number: ATM\_Forum/97-0424

\*\*\*\*\*

Title: Guaranteed Rate for Improving TCP Performance on UBR+ over Terrestrial and Satellite Networks.

\*\*\*\*\*

Abstract: We analyze the effect of providing a guaranteed rate to the UBR service class on the TCP performance over UBR. We describe an enhancement to UBR+ called the guaranteed rate service. Guaranteed rate service provides a minimum rate guarantee to the UBR traffic class. We examine effect of strict priority VBR over UBR traffic and show that the UBR performance can be improved by guaranteed rate. We also present simulation results for the effect of guaranteed rate on TCP efficiency and fairness metrics for LAN, WAN and satellite networks.

\*\*\*\*\*

Source:

Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, Xiangrong Cai  
Department of CIS, The Ohio State University (and NASA)

Raj Jain is now at Washington University in Saint Louis, [jain@cse.wustl.edu](mailto:jain@cse.wustl.edu) <http://www.cse.wustl.edu/~jain/>

Seong-Cheol Kim  
Samsung Electronics Co. Ltd.  
Chung-Ang Newspaper Bldg.  
8-2, Karak-Dong, Songpa-Ku  
Seoul, Korea 138-160  
Email: [kimsc@metro.telecom.samsung.co.kr](mailto:kimsc@metro.telecom.samsung.co.kr)

Sastri Kota  
Lockheed Martin Telecommunications  
1272 Borregas Avenue,  
Bldg B/551 O/GB - 70  
Sunnyvale, CA 94089  
Email: [sastri.kota@lmco.com](mailto:sastri.kota@lmco.com)

\*\*\*\*\*

Date: April 1997

\*\*\*\*\*

Distribution: ATM Forum Technical Working Group Members (AF-TM)

\*\*\*\*\*

Notice: This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.

\*\*\*\*\*

## 1 Introduction

TCP performance over UBR+ can be degraded when high priority VBR uses up 100% of the link. Providing a rate guarantee to the UBR class can ensure a continuous flow of TCP packets in the network. The Guaranteed Rate (GR) service provides such a guarantee to the UBR service category. The guarantees provided are for the entire UBR class, and per-VC guarantees are not provided. UBR+ with Guaranteed Rate requires no additional signalling requirements or standards changes, and can be implemented on current switches that support the UBR service. Another UBR+ service called Guaranteed Rate (GR) has been proposed in [11, 12]. The Guaranteed Rate service requires per-VC rate guarantees to UBR. This is more complex to implement and could significantly increase the cost of UBR switches. The Guaranteed Rate (GR) service is intended for applications that do not need any QoS guarantees, but whose performance depends on the availability of a continuous amount of bandwidth. GR guarantees a minimum rate to the UBR applications, while maintaining the simplicity of the basic UBR service. This guarantee is made for the entire UBR class for each link in the switch. The goal of GR is to protect the UBR class from total bandwidth starvation, and provide a continuous minimum bandwidth guarantee. In the presence of high load of higher priority CBR, VBR and ABR traffic, TCP congestion control mechanisms are expected to benefit from a guaranteed minimum rate.

In this paper, we discuss the performance of TCP with UBR in the presence of higher priority traffic. We present simulation results that show how the performance of TCP over UBR can degrade in the presence of VBR, and study the behavior of TCP over UBR with GR. Simulation results on the performance of TCP over UBR with and without GR are presented.

## 2 TCP over UBR+

In this section, we describe the basic TCP congestion control mechanisms and the UBR+ drop policies. We briefly discuss our implementations of the UBR+ switch drop policies used in our simulations to optimize TCP performance over UBR.

### 2.1 TCP Congestion Control

TCP uses a window based protocol for congestion control. The sender TCP maintains a variable called congestion window (CWND) that limits the number of unacknowledged packets that can be sent. Current and proposed versions of the TCP protocol use the following three methods for congestion avoidance and control. For a detailed discussion on the TCP model and its performance over UBR+, refer to [10].

- **Slow start and congestion avoidance (Vanilla TCP).** The sender TCP detects congestion when a retransmission timeout expires. At this time, half the congestion window value is saved in Ssthresh, and CWND is set to one segment. The sender now doubles CWND every round trip time until CWND reaches Ssthresh, after which CWND is increased by one segment every round trip time. These two phases correspond to an exponential increase and a linear increase in CWND respectively. The retransmission timeout is maintained as a coarse granularity timer. As a result, even when a single packet is dropped, much time is lost waiting for the timeout to occur and then to increase the window back to Ssthresh.
- **Fast retransmit and recovery (Reno TCP).** This mechanism was developed to optimize TCP performance for isolated segment losses due to errors. If a segment is lost, the data receiver sends duplicate ACKs for each out of sequence segment it receives. The sending TCP waits for three duplicate ACKs and retransmits the lost packet immediately. It then waits for half a window and sends a segment for each subsequent duplicate ACK. When the retransmitted packet is ACKed, the sending TCP sets CWND to half of its original value, and enters the congestion avoidance phase.
- **Selective acknowledgements (SACK TCP).** Fast retransmit and recovery cannot recover efficiently from multiple packet losses in the same window. Selective acknowledgements can be provided by the receiving TCP indicating to the sender the packets it has received. During the fast retransmission phase, the sender first retransmits the lost packets before sending out any new packets.

## 2.2 UBR+ Drop Policies

In [9], we examined TCP performance over UBR using various switch drop policies. These policies are:

- **Tail Drop.** This is the simplest possible drop policy, where the switch drops cells when the buffer becomes full. Tail drop typically results in poor performance of TCP over UBR.
- **Early Packet Discard (EPD).** Early Packet Discard maintains a threshold in its buffer. When the buffer occupancy exceeds the threshold, EPD drops complete new incoming packets to the buffer. EPD avoids the transmission of incomplete packets and increases TCP throughput over tail drop.
- **Selective Drop (SD) and Fair Buffer Allocation.** These schemes use per-VC accounting to maintain buffer utilizations of each active VC in the switch. When the buffer occupancy exceeds a preset threshold, complete packets are dropped from connections that are using more buffer than others. As a result greater fairness is achieved.

## 3 The UBR+ Guaranteed Rate Model

In this section we describe our implementation of the UBR+ GR model. Our ATM switch model is output buffered, where each output port has a separate buffer for each service category. Figure 1 shows our switch model. The switch supports multiple service categories as shown in the figure. Each service category is provided with a bandwidth guarantee. In our examples, we consider only two classes – VBR and UBR. VBR typically has strict priority over UBR, but with GR, UBR is guaranteed a fraction ( $=GR$ ) of the total link capacity.

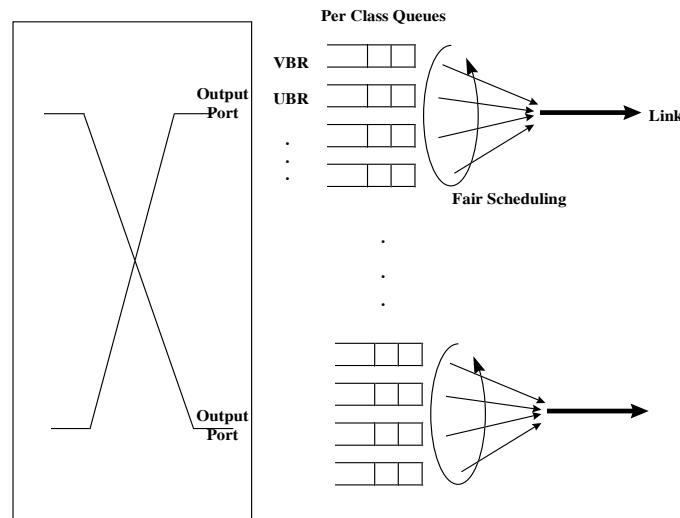


Figure 1: Switch model for UBR with GR

To enforce a GR (as a fraction of the total link capacity), we perform fair scheduling among the queues on each port. Our fair scheduling algorithm ensures that when  $GR \neq 0.0$ , the UBR class is never starved, i.e., on the average, for every  $N$  cells transmitted on to the link,  $GR \times N$  cells are from the UBR queue. This means that the VBR cells could be queued if the VBR connections are using more than  $(1-GR)$  of the link capacity. Any unused capacity by VBR is also allocated to UBR. The cell level minimum rate guarantee translates directly to a packet level guarantee for the TCP connections, because all TCP segment sizes in our simulations are the same. When transport packet sizes are different, per packet scheduling can be performed to provide frame level guarantees. The details of the scheduling algorithm will be presented in a future contribution.

Figure 2 shows the link capacity allocations for three values of GR. There is a single VBR source with an on/off burst pattern, which uses up 100% of the link capacity during the on period, and zero capacity during the off period. In the figure, the VBR on and off times are equal, so the average bandwidth requirements for VBR is 50% of the link capacity. When GR is 0, the VBR service is assigned strict priority over the UBR service. UBR is not guaranteed any rate, and must use whatever capacity is left over by the VBR source. The VBR bursts are scheduled just as they arrive and VBR cells are not queued. When  $GR = 0.1$ , 10% of the link capacity is guaranteed to the UBR service class. This 10% must be shared by all the UBR connections going through the link. In this case, if the VBR bursts may be queued in the VBR buffer to allow for UBR cells to be scheduled. The VBR bursts are thus flattened out with the VBR allocated Peak Cell Rate equal to 90% of the link capacity. Any link capacity unused by the VBR source is also available for UBR to use.

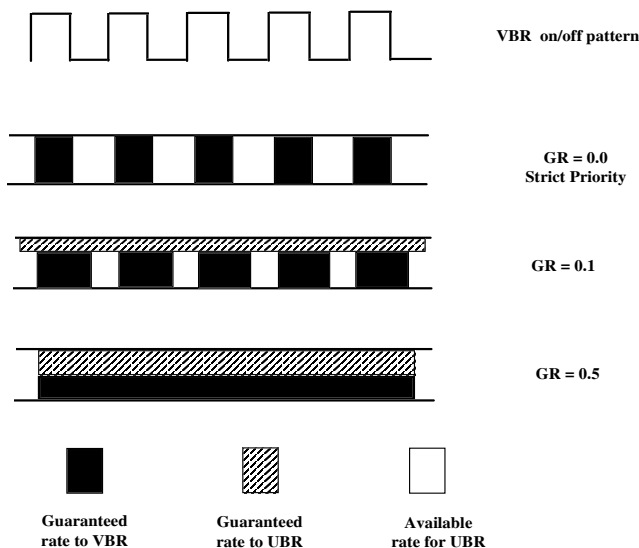


Figure 2: Link Capacity allocations for VBR and UBR with GR

When  $GR = 0.5$ , the VBR is further smoothed out so that it is now allocated a steady rate of 50% of the link capacity. On the average, the VBR queues are zero, but the on/off pattern results in temporary bursts until the burst can be cleared out.

In each of the three GR allocations, VBR uses up only 50% of the link capacity. As a result, UBR can use up to the remaining 50%. The difference between the three configurations is the way in which UBR is given the 50% capacity. With  $GR = 0$ , UBR is starved for the time VBR is using up 100% guaranteed a continuous flow of bandwidth, and is never completely starved.

In this work, we experiment with a per-port bandwidth guarantee for UBR. The study of UBR with per-VC rate guarantees is a subject of future study.

## 4 Simulation of SACK TCP over UBR+

This section presents the simulation results of the various enhancements of TCP and UBR presented in the previous sections.

### 4.1 The Simulation Model

All simulations use the N source configuration shown in figure 3. Some simulations use an additional VBR source not shown in the figure. The VBR sources is also an end to end VBR source like the other TCP connections. All sources 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 performance of TCP over UBR with bidirectional traffic is a topic of further study. The delayed acknowledgement timer is deactivated, and the receiver sends an ACK as soon as it receives a segment.

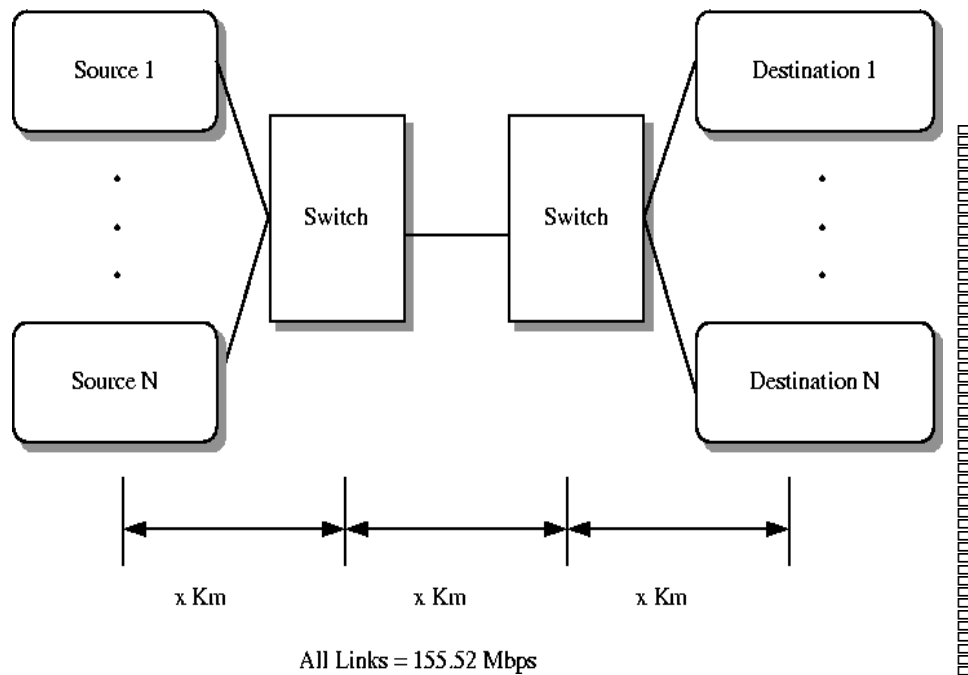


Figure 3: The N source TCP configuration

Link delays are 5 microseconds for LAN configurations and 5 milliseconds for WAN configurations. This results in a round trip propagation delay of 30 microseconds for LANs and 30 milliseconds for WANs respectively. For satellite configurations, the propagation delay between the two switches is 275 milliseconds and the distance between the TCPs and the switches is 1 km. The round trip propagation delay for satellite networks is about 540 milliseconds.

The TCP segment size is set to 512 bytes for LAN and WAN configurations. This is the common segment size used in most current TCP implementations. For satellite networks, larger segment sizes have been proposed, and 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 WAN configurations, a window of 64K bytes is not sufficient to achieve 100% utilization. We thus use the window scaling option to specify a maximum window size of 600000 Bytes. For satellite configurations, this value is further scaled up to 8704000 Bytes.

All link bandwidths are 155.52 Mbps, and Peak Cell Rate at the ATM layer is 155.52 Mbps. The duration of the simulation is 10 seconds for LANs, 20 seconds for WANs and 40 seconds for satellites. This allows for adequate round trips for the simulation to give stable results.

### 4.2 Performance Metrics

The performance of the simulation is measured at the TCP layer by the Efficiency and Fairness as defined below.

$$\text{Efficiency} = (\text{Sum of TCP throughputs}) / (\text{Maximum possible TCP throughput})$$

TCP throughput is measured at the destination TCP layer as the total number of bytes delivered to the application divided by the simulation time. This is divided by the maximum possible throughput attainable by TCP. With 512

Table 1: SACK TCP with VBR (strict priority) : Efficiency

Config-uration	Number of Sources	Buffer (cells)	VBR period (ms)	UBR	EPD	Selective Drop
LAN	5	1000	300	0.71	0.88	0.98
LAN	5	3000	300	0.83	0.91	0.92
LAN	5	1000	100	0.89	0.97	0.95
LAN	5	3000	100	0.96	0.95	0.96
LAN	5	1000	50	0.97	0.93	0.93
LAN	5	3000	50	0.95	0.97	0.97
WAN	5	12000	300	0.42	0.43	0.61
WAN	5	36000	300	0.55	0.52	0.96
WAN	5	12000	100	0.72	0.58	0.70
WAN	5	36000	100	0.95	0.97	0.97
WAN	5	12000	50	0.97	0.65	0.73
WAN	5	36000	50	0.97	0.98	0.98

bytes of TCP data in each segment, 20 bytes of TCP header, 20 bytes of IP header, 8 bytes of LLC header, and 8 bytes of AAL5 trailer are added. This results in a net possible throughput of 80.5% of the ATM layer throughput for UBR. Without VBR, the the maximum possible throughput is 125.2 Mbps on a 155.52 Mbps link. When a VBR source uses up 50% of the capacity, then the maximum possible TCP throughput reduces to 80.5% of 50% of 155.52 Mbps. This evaluates to about 63 Mbps.

$$\text{Fairness Index} = (\sum x_i)^2 / (n \times \sum x_i^2)$$

Where  $x_i$  = throughput of the  $i$ th TCP source, and  $n$  is the number of TCP sources

## 5 Simulation Results

When higher priority VBR traffic is present in the network, TCP over UBR may get considerably lower link capacity than without VBR. Moreover, the presence of VBR traffic could result in the starvation of UBR traffic for periods of time for which VBR uses up the entire link capacity. When VBR has strict priority over UBR, TCP (over UBR) traffic is transmitted in bursts and the round trip time estimates for the TCP connection are highly variable. An underestimation of the RTT is likely to cause a false timeout in the TCP indicating congestion even though the TCP packet is queued behind a VBR burst. An overestimation of the RTT may result in much time being wasted waiting for a timeout when a packet is dropped due to congestion.

### 5.1 SACK TCP over UBR+ with strict priority VBR background

The effect of UBR starvation is seen in tables 1 and 2. In this set of simulations, we used five source LAN and WAN configurations with SACK TCP. SACK TCP was chosen because it provides best performance for TCP over UBR+ [10]. 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. VBR was given strict priority over UBR, i.e., GR for UBR was 0. From the tables we can see that longer VBR bursts (for the same average VBR usage of 50%) result in lower throughput for TCP over UBR+.

Figure 4 shows the efficiency versus fairness plots for tables 1 and 2. The desirable points are those on the upper right corners of the plots, i.e., those with high efficiency and fairness values. For the WAN configuration, the upper right corner points are those from the lower VBR on/off frequencies (50 and 100 ms). With 300 ms VBR, TCP performance for WANs is poor. This is because, the VBR burst time is of the order of the TCP timeout value (2 to 3 ticks of 100 ms each). As a result the TCP source is starved long enough that a retransmission timeout occurs.

Table 2: SACK TCP with VBR (strict priority) : Fairness

Config-uration	Number of Sources	Buffer (cells)	VBR period (ms)	UBR	EPD Drop	Selective
LAN	5	1000	300	0.21	0.20	0.20
LAN	5	3000	300	0.95	0.99	0.99
LAN	5	1000	100	0.21	0.20	0.99
LAN	5	3000	100	0.91	0.93	0.96
LAN	5	1000	50	0.20	0.21	0.96
LAN	5	3000	50	0.93	0.99	1.00
WAN	5	12000	300	0.99	0.97	0.82
WAN	5	36000	300	0.88	0.97	0.63
WAN	5	12000	100	0.99	0.96	0.93
WAN	5	36000	100	1.00	0.88	0.89
WAN	5	12000	50	0.92	0.98	0.97
WAN	5	36000	50	1.00	0.97	0.80

Much time (several roundtrips of at least 30 ms each) is then wasted in recovering from the timeout during the slow start phase. This causes poor utilization of the link and lower efficiency values. When VBR on/off times are smaller compared to the retransmission value, the UBR delay is not enough to result in a TCP timeout and higher throughput results.

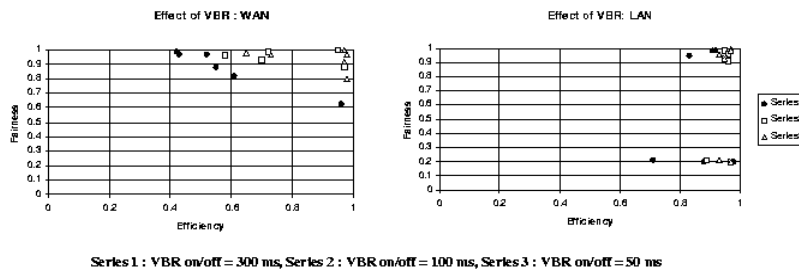


Figure 4: Variable VBR frequencies of UBR+ with strict priority

For LANs, the above argument also holds, but other factors are more dominant. The LAN plot in 4 shows that the effects of the switch drop policy and the buffer size are also important. The selective drop policy significantly improves the LAN performance of TCP over UBR+. This is because the round trip time is very small, and even during the congestion avoidance phase, the recovery is very fast. The TCP timeouts are often in phase with the VBR burst times. As a result, TCP is waiting for the timer to expire, and not utilizing the link, VBR is using the link at 100% capacity. When TCP times out and starts to send segments, the congestion window increases very fast.

### 5.1.1 SACK TCP over Guaranteed Rate UBR+ with VBR background

We now present simulation results for TCP over UBR+ with various GR values. For LAN, WAN and satellite configurations, we ran simulations with the following parameters :

- Number of sources = 5, 15 for LAN and WAN. For satellite networks, we ran the same set but only for 5 sources.
- Buffer size = 1000 cells and 3000 cells for LANs, 12000 cells and 36000 cells for WANs; and 200000 cells and 600000 cells for satellites.
- Vanilla TCP (with only slow start and congestion avoidance), Reno TCP (with fast retransmit and recovery) and SACK TCP.
- Tail Drop UBR, EPD and Selective Drop.
- UBR GR = 0.5, 0.1, 0.0 of the link capacity.

The tables 3 – 9 in the list the results of the simulations. From the tables, we categorized the results in terms of the highest efficiency and fairness values. The plots in figure 5 summarize the results in the tables.

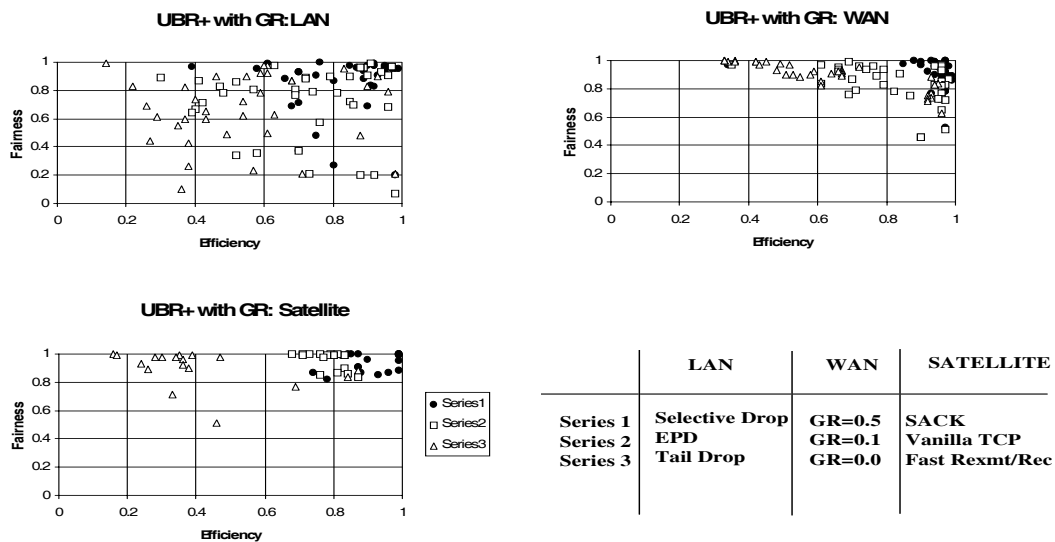


Figure 5: TCP performance over UBR+ with GR

The following observations can be made from the tables and the plots:

1. **For LANs, the dominating factor that effects the performance is the switch drop policy.** Series 1 in the figure represents the points for the selective drop policy. Clearly, selective drop 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. As a result, TCP can easily overload the switch buffers. This makes buffer management very important for LANs.
2. **For WANs, the dominating factor is the GR, and a GR of 0 hurts the TCP performance.** GR values of 0.5 and 0.1 produce the highest throughput and efficiency values. A constant amount of bandwidth provided by GR ensures that TCP keeps receiving ACKs from the destination. This reduces the variation in the round trip times. Consequently, TCP is less likely to timeout. Buffer management policies do have an



impact on TCP performance over WANs, but the effect is less than in LANs. This is because the buffer sizes of WAN 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.

3. **For satellite 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 satellite 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 over satellite networks 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 second, and so a variation of the RTT by 300 milliseconds can be tolerated by the TCP. GR may have more impact on satellite networks in cases where UBR is starved for times larger than the round trip time of the connection.

## 6 Summary

In this paper we examined the effect of higher priority VBR traffic on the performance of TCP over UBR+. Several factors can effect the performance of TCP over UBR in the presence of higher priority VBR traffic. These factors include:

- The propagation delay of the TCP connection.
- The TCP congestion control mechanisms.
- The UBR switch drop policies.
- The Guaranteed Rate provided to UBR.

For large propagation delays, end-to-end congestion control is the most important factor. For small propagation delays, the limited switch buffers makes buffer management very important. A minimum bandwidth guarantee improves TCP performance over UBR when the TCP connection may be starved for periods longer than the round trip propagation delay. The minimum bandwidth scheme explored here provides a minimum rate to the entire UBR class on the link. Per-VC GR mechanisms are an area of future study.

## References

- [1] Allyn Romanov, Sally Floyd, "Dynamics of TCP Traffic over ATM Networks," IEEE JSAC, May 1995.
- [2] ATM Forum, "ATM Traffic Management Specification Version 4.0," April 1996, <ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps>
- [3] Chien Fang, Arthur Lin, "On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme," ATM FORUM 95-1645, December 1995.
- [4] Hongqing Li, Kai-Yeung Siu, and Hong-Ti Tzeng, "TCP over ATM with ABR service versus UBR+EPD service," ATM FORUM 95-0718, June 1995.
- [5] 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.
- [6] Hongqing Li, Kai-Yeung Siu, Hong-Yi Tzeng, Brian Hang, Wai Yang, " Issues in TCP over ATM," ATM FORUM 95-0503, April 1995.
- [7] J. Jaffe, "Bottleneck Flow Control," IEEE Transactions on Communications, Vol. COM-29, No. 7, pp. 954-962.
- [8] Juha Heinanen, and Kalevi Kilkki, "A fair buffer allocation scheme," Unpublished Manuscript.

Table 3: TCP with VBR (300ms on/off) over UBR+ with GR : Efficiency for LAN

Config- uration	Number of Sources	Buffer (cells)	TCP	GR	UBR	EPD	Selective Drop
LAN	5	1000	SACK	0.5	0.26	0.85	0.96
LAN	5	1000	SACK	0.1	0.98	0.57	0.75
LAN	5	1000	SACK	0.0	0.71	0.88	0.98
LAN	5	3000	SACK	0.5	0.96	0.97	0.95
LAN	5	3000	SACK	0.1	0.93	0.89	0.99
LAN	5	3000	SACK	0.0	0.83	0.91	0.92
LAN	15	1000	SACK	0.5	0.38	0.74	0.92
LAN	15	1000	SACK	0.1	0.49	0.76	0.91
LAN	15	1000	SACK	0.0	0.57	0.98	0.90
LAN	15	3000	SACK	0.5	0.90	0.96	0.92
LAN	15	3000	SACK	0.1	0.61	0.94	0.96
LAN	15	3000	SACK	0.0	0.43	0.86	0.95
LAN	5	1000	Reno	0.5	0.22	0.30	0.61
LAN	5	1000	Reno	0.1	0.37	0.41	0.66
LAN	5	1000	Reno	0.0	0.14	0.92	0.39
LAN	5	3000	Reno	0.5	0.60	0.69	0.76
LAN	5	3000	Reno	0.1	0.55	0.79	0.93
LAN	5	3000	Reno	0.0	0.59	0.72	0.92
LAN	15	1000	Reno	0.5	0.43	0.52	0.70
LAN	15	1000	Reno	0.1	0.35	0.48	0.68
LAN	15	1000	Reno	0.0	0.29	0.40	0.70
LAN	15	3000	Reno	0.5	0.68	0.88	0.95
LAN	15	3000	Reno	0.1	0.63	0.81	0.97
LAN	15	3000	Reno	0.0	0.54	0.69	0.89
LAN	5	1000	Vanilla	0.5	0.46	0.47	0.58
LAN	5	1000	Vanilla	0.1	0.40	0.58	0.70
LAN	5	1000	Vanilla	0.0	0.27	0.73	0.80
LAN	5	3000	Vanilla	0.5	0.88	0.72	0.87
LAN	5	3000	Vanilla	0.1	0.61	0.63	0.90
LAN	5	3000	Vanilla	0.0	0.61	0.88	0.85
LAN	15	1000	Vanilla	0.5	0.59	0.42	0.80
LAN	15	1000	Vanilla	0.1	0.38	0.52	0.70
LAN	15	1000	Vanilla	0.0	0.36	0.39	0.75
LAN	15	3000	Vanilla	0.5	0.68	0.90	0.97
LAN	15	3000	Vanilla	0.1	0.54	0.96	0.98
LAN	15	3000	Vanilla	0.0	0.37	0.85	0.89

Table 4: TCP with VBR (300ms on/off) over UBR+ with GR : Efficiency for WAN

WAN	5	12000	SACK	0.5	0.95	0.93	0.94
WAN	5	12000	SACK	0.1	0.87	0.66	0.69
WAN	5	12000	SACK	0.0	0.42	0.43	0.61
WAN	5	36000	SACK	0.5	0.97	0.99	0.99
WAN	5	36000	SACK	0.1	0.96	0.98	0.96
WAN	5	36000	SACK	0.0	0.55	0.52	0.96
WAN	15	12000	SACK	0.5	0.88	0.85	0.90
WAN	15	12000	SACK	0.1	0.72	0.61	0.76
WAN	15	12000	SACK	0.0	0.64	0.48	0.58
WAN	15	36000	SACK	0.5	0.96	0.95	0.97
WAN	15	36000	SACK	0.1	0.95	0.94	0.97
WAN	15	36000	SACK	0.0	0.93	0.72	0.95
WAN	5	12000	Reno	0.5	0.93	0.96	0.94
WAN	5	12000	Reno	0.1	0.61	0.79	0.71
WAN	5	12000	Reno	0.0	0.34	0.45	0.33
WAN	5	36000	Reno	0.5	0.97	0.97	0.93
WAN	5	36000	Reno	0.1	0.90	0.96	0.75
WAN	5	36000	Reno	0.0	0.33	0.92	0.33
WAN	15	12000	Reno	0.5	0.97	0.94	0.97
WAN	15	12000	Reno	0.1	0.84	0.66	0.79
WAN	15	12000	Reno	0.0	0.67	0.53	0.51
WAN	15	36000	Reno	0.5	0.97	0.97	0.98
WAN	15	36000	Reno	0.1	0.96	0.96	0.97
WAN	15	36000	Reno	0.0	0.67	0.66	0.59
WAN	5	12000	Vanilla	0.5	0.94	0.97	0.96
WAN	5	12000	Vanilla	0.1	0.82	0.70	0.69
WAN	5	12000	Vanilla	0.0	0.49	0.36	0.42
WAN	5	36000	Vanilla	0.5	0.97	0.97	0.97
WAN	5	36000	Vanilla	0.1	0.96	0.90	0.94
WAN	5	36000	Vanilla	0.0	0.92	0.33	0.92
WAN	15	12000	Vanilla	0.5	0.90	0.92	0.96
WAN	15	12000	Vanilla	0.1	0.77	0.66	0.74
WAN	15	12000	Vanilla	0.0	0.67	0.61	0.67
WAN	15	36000	Vanilla	0.5	0.98	0.97	0.97
WAN	15	36000	Vanilla	0.1	0.96	0.96	0.97
WAN	15	36000	Vanilla	0.0	0.94	0.93	0.93

Table 5: SACK TCP with VBR (300ms on/off) over UBR+ with GR : Fairness for LAN

Config- uration	Number of Sources	Buffer (cells)	TCP	GR	UBR	EPD	Selective Drop
LAN	5	1000	SACK	0.5	0.69	0.90	0.97
LAN	5	1000	SACK	0.1	0.21	0.81	0.91
LAN	5	1000	SACK	0.0	0.21	0.20	0.20
LAN	5	3000	SACK	0.5	0.79	0.97	0.94
LAN	5	3000	SACK	0.1	0.90	0.96	0.95
LAN	5	3000	SACK	0.0	0.95	0.99	0.99
LAN	15	1000	SACK	0.5	0.43	0.79	0.83
LAN	15	1000	SACK	0.1	0.49	0.57	0.84
LAN	15	1000	SACK	0.0	0.23	0.07	0.69
LAN	15	3000	SACK	0.5	0.83	0.91	0.98
LAN	15	3000	SACK	0.1	0.50	0.93	0.91
LAN	15	3000	SACK	0.0	0.65	0.70	0.96
LAN	5	1000	Reno	0.5	0.83	0.89	0.99
LAN	5	1000	Reno	0.1	0.60	0.87	0.88
LAN	5	1000	Reno	0.0	0.99	0.20	0.97
LAN	5	3000	Reno	0.5	0.98	0.81	1.00
LAN	5	3000	Reno	0.1	0.90	0.90	0.91
LAN	5	3000	Reno	0.0	0.92	0.89	0.98
LAN	15	1000	Reno	0.5	0.60	0.86	0.93
LAN	15	1000	Reno	0.1	0.55	0.78	0.69
LAN	15	1000	Reno	0.0	0.61	0.67	0.37
LAN	15	3000	Reno	0.5	0.87	0.96	0.98
LAN	15	3000	Reno	0.1	0.63	0.78	0.95
LAN	15	3000	Reno	0.0	0.72	0.77	0.94
LAN	5	1000	Vanilla	0.5	0.90	0.83	0.95
LAN	5	1000	Vanilla	0.1	0.74	0.36	0.93
LAN	5	1000	Vanilla	0.0	0.44	0.21	0.27
LAN	5	3000	Vanilla	0.5	0.48	0.88	0.96
LAN	5	3000	Vanilla	0.1	0.92	0.98	0.98
LAN	5	3000	Vanilla	0.0	0.98	0.96	0.98
LAN	15	1000	Vanilla	0.5	0.78	0.71	0.87
LAN	15	1000	Vanilla	0.1	0.26	0.34	0.71
LAN	15	1000	Vanilla	0.0	0.10	0.64	0.48
LAN	15	3000	Vanilla	0.5	0.87	0.91	0.96
LAN	15	3000	Vanilla	0.1	0.62	0.68	0.95
LAN	15	3000	Vanilla	0.0	0.82	0.72	0.88

Table 6: SACK TCP with VBR (300ms on/off) over UBR+ with GR : Fairness for WAN

WAN	5	12000	SACK	0.5	0.95	1.00	0.99
WAN	5	12000	SACK	0.1	0.75	0.92	0.99
WAN	5	12000	SACK	0.0	0.99	0.97	0.82
WAN	5	36000	SACK	0.5	0.95	0.86	0.89
WAN	5	36000	SACK	0.1	0.96	0.87	0.77
WAN	5	36000	SACK	0.0	0.88	0.97	0.63
WAN	15	12000	SACK	0.5	1.00	0.98	0.99
WAN	15	12000	SACK	0.1	0.96	0.97	0.96
WAN	15	12000	SACK	0.0	0.91	0.93	0.90
WAN	15	36000	SACK	0.5	0.92	0.98	0.96
WAN	15	36000	SACK	0.1	0.73	0.96	0.83
WAN	15	36000	SACK	0.0	0.74	0.95	0.84
WAN	5	12000	Reno	0.5	0.77	0.93	0.96
WAN	5	12000	Reno	0.1	0.84	0.94	0.79
WAN	5	12000	Reno	0.0	0.99	0.99	1.00
WAN	5	36000	Reno	0.5	0.87	1.00	0.97
WAN	5	36000	Reno	0.1	0.46	0.82	0.97
WAN	5	36000	Reno	0.0	1.00	0.71	1.00
WAN	15	12000	Reno	0.5	0.53	0.90	0.91
WAN	15	12000	Reno	0.1	0.91	0.95	0.83
WAN	15	12000	Reno	0.0	0.91	0.90	0.90
WAN	15	36000	Reno	0.5	0.90	0.79	0.96
WAN	15	36000	Reno	0.1	0.65	0.73	0.51
WAN	15	36000	Reno	0.0	0.89	0.92	0.92
WAN	5	12000	Vanilla	0.5	0.99	0.78	0.89
WAN	5	12000	Vanilla	0.1	0.78	0.87	0.76
WAN	5	12000	Vanilla	0.0	0.98	0.99	0.99
WAN	5	36000	Vanilla	0.5	1.00	0.78	0.98
WAN	5	36000	Vanilla	0.1	0.93	0.46	0.83
WAN	5	36000	Vanilla	0.0	0.75	1.00	0.73
WAN	15	12000	Vanilla	0.5	0.97	0.92	0.95
WAN	15	12000	Vanilla	0.1	0.89	0.94	0.94
WAN	15	12000	Vanilla	0.0	0.93	0.85	0.92
WAN	15	36000	Vanilla	0.5	0.89	0.88	0.92
WAN	15	36000	Vanilla	0.1	0.97	0.85	0.72
WAN	15	36000	Vanilla	0.0	0.83	0.77	0.88

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

Drop Policy	TCP	Buffer	GR	Efficiency	Fairness
Selective Drop	SACK	200000	0.5	0.87	0.91
Selective Drop	SACK	200000	0.1	0.78	0.82
Selective Drop	SACK	200000	0.0	0.74	0.87
Selective Drop	SACK	600000	0.5	0.99	1.00
Selective Drop	SACK	600000	0.1	0.99	0.99
Selective Drop	SACK	600000	0.0	0.99	1.00
Selective Drop	Reno	200000	0.5	0.33	0.71
Selective Drop	Reno	200000	0.1	0.24	0.93
Selective Drop	Reno	200000	0.0	0.16	1.00
Selective Drop	Reno	600000	0.5	0.35	0.99
Selective Drop	Reno	600000	0.1	0.39	0.99
Selective Drop	Reno	600000	0.0	0.30	0.98
Selective Drop	Vanilla	200000	0.5	0.83	0.90
Selective Drop	Vanilla	200000	0.1	0.71	0.99
Selective Drop	Vanilla	200000	0.0	0.81	0.87
Selective Drop	Vanilla	600000	0.5	0.79	1.00
Selective Drop	Vanilla	600000	0.1	0.80	0.99
Selective Drop	Vanilla	600000	0.0	0.76	1.00

Table 8: TCP with VBR (300ms on/off) over UBR+ with GR : Satellite

Drop Policy	TCP	Buffer	GR	Efficiency	Fairness
Early Packet Discard	SACK	200000	0.5	0.84	1.00
Early Packet Discard	SACK	200000	0.1	0.88	0.87
Early Packet Discard	SACK	200000	0.0	0.82	0.99
Early Packet Discard	SACK	600000	0.5	0.99	0.95
Early Packet Discard	SACK	600000	0.1	0.99	0.88
Early Packet Discard	SACK	600000	0.0	0.99	1.00
Early Packet Discard	Reno	200000	0.5	0.46	0.51
Early Packet Discard	Reno	200000	0.1	0.26	0.89
Early Packet Discard	Reno	200000	0.0	0.17	0.99
Early Packet Discard	Reno	600000	0.5	0.36	0.96
Early Packet Discard	Reno	600000	0.1	0.34	0.98
Early Packet Discard	Reno	600000	0.0	0.28	0.98
Early Packet Discard	Vanilla	200000	0.5	0.71	1.00
Early Packet Discard	Vanilla	200000	0.1	0.76	0.85
Early Packet Discard	Vanilla	200000	0.0	0.68	1.00
Early Packet Discard	Vanilla	600000	0.5	0.78	0.99
Early Packet Discard	Vanilla	600000	0.1	0.80	0.99
Early Packet Discard	Vanilla	600000	0.0	0.77	0.98

Table 9: TCP with VBR (300ms on/off) over UBR+ with GR : Satellite

Drop Policy	TCP	Buffer	GR	Efficiency	Fairness
UBR	SACK	200000	0.5	0.87	0.91
UBR	SACK	200000	0.1	0.87	1.00
UBR	SACK	200000	0.0	0.85	1.00
UBR	SACK	600000	0.5	0.93	0.85
UBR	SACK	600000	0.1	0.96	0.87
UBR	SACK	600000	0.0	0.90	0.96
UBR	Reno	200000	0.5	0.87	0.88
UBR	Reno	200000	0.1	0.36	0.92
UBR	Reno	200000	0.0	0.38	0.9
UBR	Reno	600000	0.5	0.84	0.84
UBR	Reno	600000	0.1	0.69	0.77
UBR	Reno	600000	0.0	0.47	0.98
UBR	Vanilla	200000	0.5	0.87	0.84
UBR	Vanilla	200000	0.1	0.73	1.00
UBR	Vanilla	200000	0.0	0.84	0.86
UBR	Vanilla	600000	0.5	0.83	0.99
UBR	Vanilla	600000	0.1	0.83	0.99
UBR	Vanilla	600000	0.0	0.81	1.00

- [9] 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.
- [10] R. Goyal, R. Jain et.al., "Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks," ATM FORUM 97-0423, April 1997.
- 1
- [11] Roch Guerin, and Juha Heinanen, "UBR+ Service Category Definition," ATM FORUM 96-1598, December 1996.
- [12] Roch Guerin, and Juha Heinanen, "UBR+ Enhancements," ATM FORUM 97-0015, February 1997.
- [13] Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, Fang Lu and Saragur Srinidhi, "Performance of TCP/IP over ABR," Proc. IEEE Globecom'96, November 1996.
- [14] Shivkumar Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy and Seong-Cheol Kim, "Performance of TCP over ABR on ATM backbone and with various VBR background traffic patterns," Proc. ICC'97, June 1997.
- [15] Stephen Keung, Kai-Yeung Siu, "Degradation in TCP Performance under Cell Loss," ATM FORUM 94-0490, April 1994.
- [16] Tim Dwight, "Guidelines for the Simulation of TCP/IP over ATM," ATM FORUM 95-0077r1, March 1995.
- [17] V. Jacobson, "Congestion Avoidance and Control," Proceedings of the SIGCOMM'88 Symposium, pp. 314-32, August 1988.
- [18] V. Jacobson, R. Braden, "TCP Extensions for Long-Delay Paths," Internet RFC 1072, October 1988.
- [19] V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance," Internet RFC 1323, May 1992.
- [20] Kevin Fall, Sally Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP,"
- [21] Sally Floyd, "Issues of TCP with SACK,"

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

- [22] M. Mathis, J. Madhavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgement Options," Internet RFC 2018, October 1996.
- [23] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," Internet RFC 2001, January 1997.
- [24] Janey C. Hoe, "Start-up Dynamics of TCP's Congestion Control and Avoidance Schemes," MS Thesis, Massachusetts Institute of Technology, June 1995.
- [25] Mark Allman, Chris Hayes, Hans Kruse, Shawn Ostermann, "TCP Performance over Satellite Links," Proc. 5th International Conference on Telecommunications Systems, 1997.