Performance of TCP over ABR on ATM backbone and with various VBR background traffic patterns

Shivkumar Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy and Seong-Cheol Kim*
Computer and Information Sciences, The Ohio State University
395 Dreese Labs, 2015 Neil Avenue, Columbus, OH 43210-1277, U.S.A.
Email: {*shivkuma, jain, goyal, fahmy*}@cis.ohio-state.edu

Abstract

We extend our earlier studies of buffer requirements of TCP over ABR [9, 10, 11] in two directions. First, we study the performance of TCP over ABR in an ATM backbone. We find that the TCP queues are at the edge router and not inside the ATM network. The edge router requires buffering equal to the sum of the receiver window sizes of the participating TCP connections. Second, we study the performance when ABR capacity is variable due to the effect of various patterns of VBR background traffic. The key factors in this study are the VBR traffic pattern, ABR feedback delays and the sensitivity of the ABR switch scheme to variance. We present our experiences with refining the ERICA+ switch scheme [7] to handle these conditions.

1 Introduction

With the proliference of multimedia traffic over the Internet, several technologies capable of handling such traffic efficiently are competing to replace various backbones and subnetworks of the Internet. The Asynchronous Transfer Mode (ATM) technology which has been designed specifically to support integration of data, voice, and video applications is one of the key technologies in this competition.

ATM networks provide multiple classes of service to support the Quality of Service (QoS) requirements of different applications [1]. Of these classes, the Available Bit Rate (ABR) service class has been developed for the the fair and efficient support of data applications. Along with the Unspecified Bit Rate (UBR) class, it uses the link capacity left over after the higher priority classes (like the Variable Bit Rate (VBR) class) have been serviced.

The performance of Internet traffic using TCP/IP and running over ATM using the ABR and UBR services has been the focus of several recent studies [3, 4, 5, 9, 10, 11, 12]. In this paper, we first study the buffering issues in using the ABR service to transport TCP traffic over ATM backbones. Second, we study the performance in when the available capacity for ABR is highly variant due to the VBR background traffic. We present our experiences in refining the ERICA+ switch scheme [7] to handle such conditions.

2 TCP Behavior over ABR

TCP provides a reliable transfer of data using a window-based flow and error control algorithm [2]. When TCP uses ABR, the TCP window-based control runs on top of the ABR rate-based control. ABR has a closed-loop traffic management mechanism where the ABR switch scheme typically measures the ABR load and capacity in one direction and gives feedback in resource management (RM) cells traveling in the reverse direction. The ABR source end systems respond to the feedback and generates RM cells as described in [6]. The TCP traffic appears bursty at the ATM layer [10], i.e., it has active periods when there is data to send and idle periods when there is no data to send (it is waiting for acks). This behavior results in some effects seen at the ATM switch:

a) **Out-of-phase effect:** No ABR load and ABR sources are seen in the forward direction while ABR sources and RM cells are seen in the reverse direction.

b) **Clustering effect:** Since cells from TCP connections come in clusters, some sources may not be seen (and considered inactive) in a given interval of time.

Due to these effects, switches may make errors in measuring quantities which they use to calculate feedback. However, these effects reduce when the ABR feedback control becomes more effective. This occurs when the TCP source congestion windows grow and the network path is completely filled by TCP traffic. If the buffers are not sufficient to hold the excess traffic during this phase, cells are lost and TCP performance degrades. We show in [8] that, since TCP has a built in congestion avoidance mechanism, it does not lose too many cells. However, the throughput is low because of the time lost during TCP timeout and retransmission. Under these conditions, a smaller TCP timer granularity and intelligent switch drop policies help improve performance of TCP. Several authors have studied the performance of TCP

---

*Seong-Cheol Kim is with Samsung Electronics Co. Ltd., Korea

0-7803-3925-8/97 $10.00

over ABR and UBR services under lossy conditions using different switch drop policies [3, 4, 5, 12].

Note that TCP achieves maximum throughput when there sufficient buffer to guarantee no packet loss [9, 10, 11]. To achieve zero-loss, ABR service requires switch buffering which is only a small multiple of the round trip time (RTT) and the feedback delay. The buffering required depends heavily upon the switch scheme used.

Once the ATM source rates are controlled, the queues build up at the sources, and not at the switches. In effect, the ABR queues are pushed to the edge of the ATM network. In an ATM backbone network, the source is the edge router. In this paper, we quantify the buffering requirement at the edge router and discuss related issues.

The introduction of VBR traffic makes the ABR capacity variable resulting in more variance at the switch. We study the effect of using different VBR background patterns, the feedback delay, and the switch scheme used. We present our experiences with refining the ERICA+ switch scheme [7] to handle such conditions.

3 The ERICA and ERICA+ Switch Schemes

In this section, we present a brief overview of the ERICA and ERICA+ switch algorithm. More details can be found in reference [7].

Explicit Rate Indication for Congestion Avoidance (ERICA) is a simple switch scheme that allocates bandwidth fairly with a fast response. The scheme consists of using a Target Utilization of, say, 90%. The Target Rate is then set at:

Target Rate = Target Utilization × Link Bandwidth

Since VBR and CBR are serviced first, a simple way of calculating bandwidth available for ABR service class is:

Measured ABR Capacity = Target Rate - VBR Bandwidth Usage - CBR Bandwidth Usage

An overload factor (z) which is the ratio the input rate and the ABR capacity is measured regularly:

Overload Factor = Input Rate / ABR Capacity

In addition to the overload factor, the switches also measure the number of active VCs (Na) and compute a fairshare as follows. The fairshare is the minimum allocation given to any active VC.

Fairshare = ABR Capacity/ Number of Active VCs

For each VC, a share is computed based on the overload factor and the VC's current cell rate. This term is used to achieve efficiency and full link utilization, in cases where some sources do not utilize their fairshare allocations:

VC's Share = VC's Current Cell Rate / Overload Factor

The VC is given the maximum of its share as computed above or the fairshare.

ER for VC = max (Fairshare, VC's Share)

The explicit rate (ER) in the RM cell is reduced if ER for VC as computed above is less:

ER in Cell = min (ER in Cell, ER for the VC)

This simple algorithm has several desirable properties including fast response time, low queue length, and simplicity. There are other steps in the algorithm which ensure fairness and reduce transient overloads [7], but the above outline is sufficient for the discussion in this paper.

The ERICA algorithm uses two key parameters: target utilization and averaging interval length. The algorithm measures the overload factor and number of active sources over successive averaging intervals and tries to achieve a link utilization equal to the target. The averaging intervals end either after the specified length or after a specified number of cells have been received, whichever happens first.

In the simulations reported here, the target utilization is set at 90%, and the averaging interval length defaults to 1 ms or 100 ABR input cells, represented as the tuple (1 ms, 100 cells). However, our study of source end-system queues and implications for ATM backbone networks uses ERICA with some of the modifications suggested in Section 6.2.4 and a large averaging interval of (5 ms, 500 cells).

The ERICA+ algorithm is an extension of ERICA which uses the queueing delay as a additional metric to calculate the feedback. It scales the measured ABR capacity based on the queue length (q) information as follows:

1. *Q0 = Measured ABR Capacity × T0. (T0 is an input parameter)*

2a. $f(T_q) = \mathit{Max}\ (\mathit{QDLF},\ (a \times Q0)/((a-1) \times q + Q0))$ *for* $q > Q0$

2b. $f(T_q) = (b \times Q0)/((b-1) \times q + Q0)$ *for* $0 \leq q \leq Q0$

3. *ABR Capacity =* $f(T_q)\times$ *Measured ABR capacity*

The remaining steps of the algorithm are the same as in ERICA (starting from overload factor measurement). Note that ERICA+ eliminates the target utilization parameter (set to 1.0) and uses four new parameters: a target queueing delay (T0 = 500 microseconds), two curve parameters (a = 1.15 and b = 1.05), and a factor which limits the amount of ABR capacity allocated to drain the queues (QDLF = 0.5).

4 TCP Parameters

We use a TCP maximum segment size (MSS) of 512 bytes. The MTU size used by IP is generally 9180 bytes and so there is no segmentation caused by IP. We implemented the window scaling option so that the throughput is not limited by path length. Without the window scaling option, the maximum window size is 2^{16} bytes or 64 kB. We use a maximum receiver window of 16 × 64 kB or 1024 kB. The network consists of three links of 1000 km (maximum) each and therefore, has a maximum one-way delay of 15 ms (or 291 kB at 155 Mbps). The maximum receiver window is, thus, greater than twice the one-way delay. We use a TCP timer granularity of 100 ms. The timer is exercised only when there is packet loss.

The TCP data is encapsulated over ATM as follows. First, a set of headers and trailers are added to every TCP segment. We have 20 bytes of TCP header, 20 bytes of IP header, 8 bytes for the RFC1577 LLC/SNAP encapsulation, and 8 bytes of AAL5 information, a total of 56 bytes. Hence, every MSS of 512 bytes becomes 568 bytes of payload for transmission over ATM. This payload with padding requires 12 ATM cells of 48 data bytes each. Hence, the maximum receiver window of 1024 kB corresponds to 24576 cells over ATM.

In our simulations, we have not used the "fast retransmit and recovery" algorithms. However, the zero-loss buffer requirement is valid for fast retransmit and recovery too, since these algorithms are not exercised when there is zero-loss.

5 N Source + VBR Configuration

The "N Source + VBR" configuration shown in figure 1 has a single bottleneck link shared by the N ABR sources and possibly a VBR source. Each ABR source is a large (infinite) file transfer application using TCP. All traffic is unidirectional. All links run at 155 Mbps. The links traversed by the connections are symmetric i.e., each link on the path has the same length for all the VCs. In our simulations, N is 15 and the link lengths may assume values 1000, 500, 100 and 1 km.

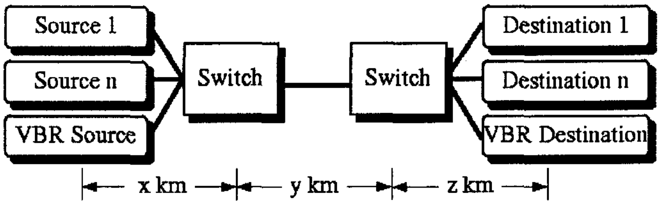

Figure 1: n Source + VBR Configuration

The individual link lengths determine the round trip time (RTT) and the feedback delay. Feedback delay is the sum of the delay for feedback from the switch to reach the source and the delay for the new load from the sources to reach the switch. It is at least twice the one-way propagation delay from the source to the switch. The feedback delay determines how quickly the feedback is conveyed to the sources and how quickly the new load is felt at the switch.

The VBR source when present is an ON-OFF source. The ON time and OFF time are defined in terms of a "duty cycle" and a "period". A pulse with a duty cycle of d and period of p has an ON time of d×p and and OFF time of (1-d)×p. Our previous results of TCP over VBR used a duty cycle of 0.5 resulting in the ON time being equal to the OFF time. Unequal ON-OFF times used in this study cause new effects that were not seen before.

The VBR starts at t = 2 ms to avoid certain initialization problems. During the ON time, the VBR source operates at its maximum amplitude. The maximum amplitude of the VBR source is 124.41 Mbps (80% of link rate). VBR is given priority at the link, i.e, if there is a VBR cell, it is scheduled for output on the link before any waiting ABR cells are scheduled.

6 Results

First, we quantify the buffer requirement at the ABR source end-system for zero-loss TCP transmission and discuss implications for backbone ATM networks. The VBR source is turned off in this study. We then introduce VBR traffic and examine the switch buffering requirement, studying the effect of varying the VBR ON-OFF periods, the ABR feedback delay and the ABR switch scheme.

6.1 TCP Performance over ATM Backbone Networks

The ATM source buffer requirement is derived by examining the maximum queues at the source when TCP runs over ABR. We also study the performance when sufficient buffers are not provided and discuss the implications for ATM backbone networks.

6.1.1 Source End System Queues in ABR

Table 1 shows the results with a 15-source configuration with link lengths of 1000 km (there is no VBR background). The RTT is 30 ms and the feedback delay, 10 ms. We vary the size of the source end-system buffers from 100 cells to 100000 cells per VC (second column). These values are compared to the maximum receiver window size (indicated as "Win" in the table) which is 1024 kB = 24576 cells. The switch has infinite buffers and uses a modified version of the ERICA algorithm [7] including the averaging feature for the number of sources and an averaging interval of (5 ms, 500 cells) as described in Section 6.2.4.

The maximum source queue values (third column) are tabulated for every VC, while the maximum switch queue values (fourth column) are for all the VCs together. When there is no buffer overflow the maximum source queue (third column) measured in units of cells is also presented as a fraction of the maximum receiver window. The switch queues are presented as a fraction of the RTT (indicated as "RTT" in the table). The RTT for this configuration is 30 ms which corresponds to a "cell length" of 30 ms × 368 cells/ms = 11040 cells.

The last column shows the aggregate TCP throughput. The maximum possible TCP throughput in our configuration is approximately: 155.52 × (0.9 for ERICA Target Utilization) × (48/53 for ATM payload) × (512/568 for protocol headers) × (31/32 for ABR RM cell overhead) = 110.9 Mbps.

In rows 1, 2 and 3 of Table 1, the source has insufficient buffers. The maximum per-source queue is equal to the source buffer size. The buffers overflow at the source and cells are dropped. TCP then times out and retransmits the lost data. TCP performance under these conditions (of insufficient source buffers and sufficient switch buffers) is similar to its performance when the switch has insufficient buffers and the source has sufficient buffers [8].

Also observe that the switch queue reaches its maximum possible value for this configuration (1.56×RTT) given a minimum amount of per-source buffering (1000 cells = 0.04×Win). The switch buffering requirement is under 3×RTT as predicted in [9, 10, 11].

The sources however require one receiver window's worth of buffering per VC to avoid cell loss. This hypothesis is substantiated by row 4 of Table 1 which shows that the maximum per-source queue is 23901 cells = 0.97×Win. The remaining cells (0.03×Win) are traversing the links inside the ATM network. The switch queues are close to zero because the sources are *rate-limited* by the ABR mechanism [8]. The TCP throughput (110.9 Mbps) is the maximum possible given this configuration, scheme and parameters.

The total buffering required for N sources is the sum of the N receiver windows. Note that this is the same as the switch buffer requirement for UBR [5]. In other words, the ABR and UBR services differ in whether the sum of the receiver windows' worth of queues is seen at the source or at the switch.

6.1.2 Implications for ATM Backbone Networks

If the ABR service is used end-to-end, then the TCP source and destination are directly connected to the ATM network. The source can directly flow control the TCP source. As a result, the TCP data stays in the disk and is not queued in the end-system buffers. In such cases, the end-system need not allocate large buffers. ABR is better than UBR in these end-to-end configurations since it allows TCP to scale well.

However, if the ABR service is used on a backbone ATM network, the end-systems are edge routers which are not directly connected to TCP sources. These edge routers may not be able to flow control the TCP sources except by dropping cells, or by modifying TCP headers in acknowledgements. To avoid cell loss, these routers need to provide one receiver window's worth of buffering per TCP connection. The buffering is independent of whether the TCP connections are multiplexed over a smaller number of VCs or there is a VC per TCP connection. For UBR, these buffers need to be provided inside the ATM network, while for ABR they need to be provided at the edge router. If there are insufficient buffers, cell loss occurs and TCP performance degrades. The fact that the ABR service pushes the congestion to the edges of the ATM network while UBR service pushes it inside is an important benifit of ABR for service providers. In general, UBR service requires more buffering in the switches than the ABR service.

6.2 Performance of TCP over ABR with VBR Background

We now continue our study of ABR switch buffering by introducing VBR traffic in addition to the 15 ABR sources. All link lengths are 1000km. The RTT is 30 ms and the feedback delay is 10 ms.

We use the ERICA+ algorithm [7] in our results. The ERICA+ algorithm aims to achieve 100% link utilization in the steady state and a target queueing delay at the switch. Although we had invented ERICA+ to allow full utilization of expensive links, we found that it is helpful in controlling queues and providing stability in cases with high variance in ABR demand and capacity. Specifically, in the presence of TCP over ABR with highly variant VBR background, the target queueing delay is never achieved. However, since the ABR capacity is scaled as a function of queue length, the queue maximum can be controlled while the link utilization remains high.

Table 2: Effect of VBR ON-OFF Times

#	Duty Cycle (d) (ms)	Period (p) (ms)	Maximum Switch Q (cells)
1.	0.95	100	2588 (0.23×RTT)
2.	0.8	100	5217 (0.47×RTT)
3.	0.7	100	5688 (0.52×RTT)
4.	0.95	10	2709 (0.25×RTT)
5.	0.8	10	Unbounded
6.	0.7	10	Unbounded
7.	0.95	1	2589 (0.23×RTT)
8.	0.8	1	4077 (0.37×RTT)
9.	0.7	1	2928 (0.26×RTT)

Table 2 shows the results of a 3×3 full-factorial experimental design used to identify the problem space with VBR background traffic. We vary the two VBR model parameters: the duty cycle (d) and the period (p). Recall that, with parameters d and p, the VBR ON time is d×p and the VBR OFF time is d×(1−p). Each parameter assumes three values. The duty cycle assumes values 0.95, 0.8 and 0.7 while the period may be 100 ms (large), 10 ms (medium) and 1 ms (small).

The maximum switch queue is also expressed as a fraction of the round trip time (30 ms = 30 ms × 368 cells/ms = 11040 cells).

Table 1: Source Queues in ABR

#	Source Buffer (cells)	Max Source Q (cells)	Max Switch Q (cells)	Total Throughput
1.	100 (< Win)	> 100 (overflow)	8624 (0.78×RTT)	73.27 Mbps
2.	1000 (< Win)	> 1000 (overflow)	17171 (1.56×RTT)	83.79 Mbps
3.	10000 (< Win)	> 10000 (overflow)	17171 (1.56×RTT)	95.48 Mbps
4.	100000 (> Win)	23901 (0.97×Win)	17171 (1.56×RTT)	110.90 Mbps

6.2.1 Effect of VBR ON-OFF Times

Rows 1, 2 and 3 of Table 2 characterize large ON-OFF times (low frequency VBR). Observe that the (maximum) queues are small fractions of the RTT. The queues which build up during the ON times are drained out during the OFF times. Given these conditions, VBR may add at most one RTT worth of queues. ERICA+ further controls the queues to small values.

Rows 4, 5 and 6 of Table 2 characterize medium ON-OFF times. We observe that rows 5 and 6 have unbounded (divergent) queues. The effect of the ON-OFF time on the divergence is explained as follows.

During the OFF time the switch experiences underload and allocates high rates to sources. The duration of the OFF time along with the feedback delay (see next section) determines how long such high rate feedback is given to sources. In the worst case, the ABR load at the switch is maximum whenever the VBR source is ON to create the largest backlogs. On the other hand, the VBR OFF times also allow the ABR queues to be drained out, since the switch is underloaded during these times. Larger OFF times may allow the queues to be completely drained before the next ON time. The queues will grow unboundedly (i.e., diverge) if the queue backlogs accumulated after ON and OFF times never get cleared.

Rows 7, 8 and 9 of Table 2 characterize small ON-OFF times. Observe again that the queues are small fractions of the round trip time. Since the OFF times are small, the switch does not have enough time to allocate high rates. Since the ON times are small, the queues do not build up significantly in one ON-OFF cycle. On the other hand, the frequency of the VBR is high. This means that the VBR changes much faster than the time required for sources to respond to feedback. In these cases, ERICA+ controls the queues to small values.

6.2.2 Effect of Feedback Delays

Another factor which interacts with the VBR ON-OFF periods is the feedback delay. We saw that one of the reasons for the divergent queues was that switches could allocate high rates during the VBR OFF times. The feedback delay is important in two ways. First, ***the maximum time for which the switch may allocate high rates is the minimum of the feedback delay and the VBR OFF-time.*** This is because, the load due to the high rate feedback is seen at the switch within one feedback delay. Second, when the switch is overloaded, ***it takes at least one feedback delay to reduce the rates of the sources.***

Table 3: Effect of Feedback Delay

#	Feedback Delay(ms)	RTT (ms)	Max Switch Q (cells)
1.	1 ms	3 ms	4176 (0.4×RTT)
2.	5 ms	15 ms	Unbounded
3.	10 ms	30 ms	Unbounded

The experiments shown in Table 2 have a long feedback delay (10 ms). The long feedback delay allows the switch to allocate high rates for the entire duration of the VBR OFF time. Further, when the switch is overloaded, the sources takes 10 ms to respond to new feedback. Therefore, given appropriate value of the ON-OFF times (like in rows 5 and 6 of Table 2), the queues may diverge.

Table 3 shows the effect of varying the feedback delay and RTT. We select the divergent case (row 5, with d = 0.8 and p = 10 ms) from Table 2 and vary the feedback delay and round trip time of the configuration.

Row 1 in Table 3 shows that the queues are small when the feedback delay is 1 ms (metropolitan area network configuration). In fact, the queues will be small when the feedback delay is smaller than 1 ms (LAN configurations). In such configurations, the minimum of the OFF time (2 ms) and the feedback delay (< 1 ms) is the feedback delay. Hence, in any VBR OFF time, the switch cannot allocate high rates to sources long enough to cause queue backlogs. The new load is quickly felt at the switch and feedback is given to the sources.

Rows 2 and 3 in Table 3 have a feedback delay longer than the OFF time. This is one of the factors causing the divergence in the queues in these cases.

6.2.3 Effect of Switch Scheme

The TCP traffic makes the ABR demand variable. The VBR background makes the ABR capacity variable. In the presence of TCP and VBR, the measurements used by switch schemes are affected by the variance in ABR demand and capacity. ***The errors in switch scheme measurements are reflected in the feedback given to sources, which in turn result in switch queues.*** Switch schemes need to be robust to perform under such error-prone conditions.

As an example, consider the case when the VBR ON-OFF periods are very small (1 ms ON, 1 ms OFF). The resulting variance can be controlled by a switch scheme like ERICA+ which uses the queueing delay to calculate feedback (in addition to input rate etc). The basic ERICA algorithm does not look at the queue lengths (which may be caused by measurement errors), and cannot handle this level of variance.

Though the ERICA+ algorithm uses the queue length as a secondary metric to reduce the high allocation of rates, it has a limit on how much it can reduce the allocation. Given sufficient variance, this limit can be reached. This means that *even the minimum rate allocation by ERICA+ may cause queues to grow unboundedly.* This is one more reason for the divergent cases seen in Tables 2 and 3.

6.2.4 Improving the Robustness of ERICA+

We tackle the robustness problem by reducing the effect of variance on the switch scheme measurements in the following ways:

1. First, we reduce variance in switch scheme measurements by ***measuring quantities over longer intervals.*** Longer intervals yield averages which have less variance. However, making the intervals too long increases the response time which may result in queues.

2. Second, we ***average the measurements over several successive intervals*** using techniques similar to exponential averaging. The ERICA+ scheme uses two important measurements: the overload factor (z) which is the ratio of the input rate and the target ABR rate, and the number of active sources (N).

   The *overload factor (z)* is used by ERICA+ to divide the current cell rate of the source to give what we call the "VC share". The VC share is one of the rates which may be given as feedback to the source.

   The overload factor is affected by the *out-of-phase effect* of TCP over ABR as follows. Consider an averaging interval when no load is seen in the forward direction, whereas RM cells are seen in the reverse direction. The overload factor (z) in this case is measured as zero (because the input rate seen in the interval is zero), and we will allocate a very high rate to the sources based on this transient value of z.

   The *number of active sources (Na)* is used to calculate a minimum fairshare that any active source will get. If Na is underestimated, then the minimum fairshare will be high leading to overallocation of rates.

   Due to the *clustering effect* of TCP, there is a high probability that cells from just a few VCs may be seen in any measurement interval. This leads to an underestimate of Na, and subsequent overallocation of rates.

   We design averaging methods for the overload factor and the number of active sources as described in [7]. The averaging methods have parameters α_z and α_n respectively, which satisfy the condition: $0 \leq \alpha_N, \alpha_z \leq 1$. Though details of the methods differ, the effect of each method is roughly equivalent to the effect of increasing the measurement interval for estimating the respective quantity, and at the same time maintaining quick response to sudden overload conditions.

Table 4: Effect of Switch Scheme

#	Avging Interval (T ms, n cells)	Avging of Na on ? ($\alpha_n = 0.9$)	Avging of z on ? ($\alpha_z = 0.2$)	Max Switch Queue (cells)
1.	(1,100)	YES	YES	5223
2.	(5,500)	YES	NO	5637

3. Third, we modify the scheme to ***handle boundary conditions gracefully*** [7]. From this study, we have learnt that boundary conditions can be a common case if the variance in the network is high, and hence a robust strategy for handling them is required. Specifically, we set the number of active sources (Na) to unity if it is measured to be below unity. Our new method for averaging the overload factor (z) described in [7] does not allow z to assume values of zero or infinity, but at the same time, does not ignore outlier measured values while calculating the average.

The ERICA+ scheme with these modifications controls the ABR queues without compromising on throughput. Table 4 shows the results of representative experiments using these features.

Row 1 shows the performance with the averaging of Na and z turned on using a formerly divergent case (with d = 0.7 and p = 20 ms). Observe that the queue converges and is small. The parameter α_z is 0.2, which is roughly equivalent to increasing the averaging interval length by a factor of 5 [7]. Hence, we try the value (500 cells, 5 ms) as the averaging interval length, without the averaging of overload factor. Row 2 shows that the queue for this case also converges and is small.

7 Summary

We have presented further results on the issue of buffering requirements for TCP over ABR. The first result deals with the source end system queues, and has significance in ABR backbone configurations. Though the ABR switch buffering requirement is small, the ATM source buffering required is equal to the sum of the TCP receiver window sizes. This is the buffering required in edge routers of the ATM network.

We then study the impact of VBR background traffic on switch buffering. We find that the ON-OFF times, the feedback delays, and a switch scheme sensitive to variance in ABR load and capacity may combine to create worst case conditions where the ABR queues diverge. We enhance the ERICA+ scheme to reduce the effect of the variance and allow the convergence of the ABR queues, without compromising on the efficiency.

References

[1] ATM Forum, "ATM Traffic Management Specification Version 4.0," April 1996, ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps

[2] V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM'88*, pp. 314-32, August 1988.

[3] C. Fang, A. Lin, "On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme," *AF-TM 95-1645*,[1] December 1995.

[4] C. Fang, A. Lin, "A Simulation Study of ABR Robustness with Binary-Mode Switches: Part II," *AF-TM 95-1328R1*, October 1995.

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

[6] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, "Source Behavior for ATM ABR Traffic Management: An Explanation," *IEEE Communications Magazine*, November 1996.

[7] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, "ERICA Switch Algorithm: A Complete Description," *AF-TM 96-1172*, August 1996.

[8] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, F. Lu, and Saragur Srinidhi, "Performance of TCP/IP over ABR," *Proc. IEEE Globecom'96*, November 1996. [2]

[9] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal and S.C. Kim, "Buffer Requirements For TCP/IP Over ABR," *Proc. IEEE ATM'96 Workshop*, August 1996.

[10] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal and S.C. Kim, "Performance and Buffering Requirements of Internet Protocols over ATM ABR and UBR Services," *IEEE Communications Magazine, submitted.*

[11] S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy and S.C. Kim, "Performance of TCP/IP Using ATM ABR and UBR Services over Satellite Networks," *Proc. IEEE Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks*, October 1996.

[12] 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.

[13] Allyn Romanov, Sally Floyd, "Dynamics of TCP Traffic over ATM Networks," *IEEE JSAC*, May 1995.

---

[1] Throughout this section, AF-TM refers to ATM Forum Traffic Management sub-working group contributions.

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