Performance of TCP/IP over ABR Service on ATM Networks

Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, Fang Lu*, Saragur Srinidhi †

Computer and Information Sciences, The Ohio State University
395 Dreese, 2015 Neil Ave., Columbus, OH 43210-1277

New Address: Raj Jain, Washington University in Saint Louis,
jain@cse.wustl.edu, http://www.cse.wustl.edu/~jain

Abstract

The Available Bit Rate (ABR) service has been developed to support data applications over Asynchronous Transfer Mode (ATM). It is hence interesting to study the performance of reliable data transport protocols like the "Transport Control Protocol (TCP)" over ABR. We study the effect of running large unidirectional file transfer applications on TCP over ABR with an explicit rate algorithm (ERICA) implemented at the ATM switches. The paper shows that it is possible to get maximum TCP throughput when there are enough buffers at the switches. However, when the number of buffers is smaller, there is a large reduction in TCP throughput even though the Cell Loss Ratio (CLR) is very small. We study the effect of various factors which affect TCP throughput and fairness. These factors include the TCP timer granularity, switch buffering, ABR parameters, and the cell drop policy at the switches.

1 Introduction

ATM networks provide four classes of service: constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR), and unspecified bit rate (UBR). Data traffic in ATM is expected to be transported by the ABR service. Link bandwidth is first allocated to the VBR and CBR classes. The remaining bandwidth, if any, is given to ABR and UBR traffic. ABR mechanism rapidly allocates this bandwidth among active ABR sources. The introduction of the ABR class is expected to yield high overall link utilization and efficient and fair support of data traffic.

The ATM Forum Traffic Management group has standardized a rate-based closed-loop flow control model for the ABR class of traffic [2]. In this model, the ATM switches give feedback (explicit rate (ER) or binary (Explicit Forward Congestion Indication, EFCI)) in Resource Management (RM) cells and the sources adjust their transmission rates appropriately.

TCP is the most popular transport protocol for data transfer. It provides a reliable transfer of data using a window-based flow and error control algorithm [6]. TCP runs over IP which in turn can run over ATM. Hence, when TCP uses the ABR service, there are two control algorithms active: the TCP window-based control running on top of the ABR rate-based control. It is important to verify that the ABR control performs satisfactorily for TCP/IP traffic.

There are recent studies for TCP over the Unspecified Bit Rate (UBR) service class [3, 4, 13]. The UBR class [2] is the lowest priority class in ATM. UBR does not include flow control and hence depends upon transport layers to provide flow control. The only degree of freedom to control traffic in UBR is through buffer allocation (which includes cell drop policies). ABR has additional degrees of freedom in terms of switch schemes and source parameters. These are discussed in Section 3.

The aforementioned TCP studies also compare UBR performance with ABR using either EFCI switches [4] or explicit rate (ER) switches in local area network (LAN) topologies. Since LANs have short feedback loops, some properties of the ABR control mechanisms may not be clearly observed in LAN configurations.

This paper studies the effect of unidirectional TCP traffic over ABR with Explicit Rate (ER) switches running the ERICA algorithm [7]. We use wide area network (WAN) configurations with finite buffers at the switches. Local area networks (LANs) have short feedback loops and the ABR control is found to be effective in LAN cases. We study the effect of timer granularity, switch buffer capacity, tail drop at switches, variable capacity of the links and the effect of the source end system (SES) parameter "Transient Buffer Exposure" (TBE). The effect of fast retransmit and recovery, early packet discard [14], other TCP factors will be studied in future papers.

2 TCP Congestion Mechanisms

TCP is one of the few transport protocols that has its own congestion control mechanisms. The key TCP congestion mechanism is the so called "Slow start." TCP connections use an end-to-end flow control window to limit the number of packets that the source sends. The sender window is the minimum of the receiver window (Wrcvr) and a congestion window variable (CWND).

Whenever a TCP connection loses a packet, the source does

---

*Fang Lu is currently with Lucent Technologies, NJ.

†Saragur Srinidhi is currently with Fore Systems, India.

0-7803-3336-5/96 $5.00

not receive an acknowledgment and it times out. The source remembers the congestion window (CWND) value at which it lost the packet by setting a threshold variable SSTHRESH at half the window. More precisely, SSTHRESH is set to max{2, min{CWND/2, Wrcvr}} and CWND is set to one.

The source then retransmits the lost packet and increases its CWND by one every time a packet is acknowledged. We call this phase the "exponential increase phase" since the window when plotted as a function of time increases exponentially. This continues until the window is equal to SSTHRESH. After that, the window w is increased by $1/w$ for every packet that is acked. This is called the "linear increase phase" since the window graph as a function of time is approximately a straight line. Note that although the congestion window may increase beyond the advertised receiver window, the source window is limited by that value. when packet losses occur, the retransmission algorithm may retransmit all the packets starting from the lost packet. That is, TCP uses a go-back-N retransmission policy. The typical changes in the source window plotted against time are shown in Figure 1.

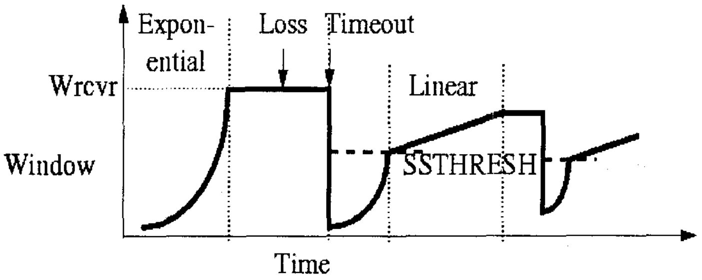


Figure 1: TCP Window vs Time using Slow Start

When there is a bursty loss due to congestion, time is lost due to timeouts and the receiver may receive duplicate packets as a result of the go-back-N retransmission strategy. This is illustrated in Figure 2. Packets 1 and 2 are lost but packets 3 and 4 make it to the destination are are stored there. After the timeout, the source sets its window to 1 and retransmits packet 1. When that packet is acknowledged, the source increases its window to 2 and sends packets 2 and 3. As soon as the destination receives packet 2, it delivers all packets upto 4 to the application and sends an ack (asking for packet 5) to the source. The 2nd copy of packet 3, which arrives a bit later is discarded at the destination since it is a duplicate.

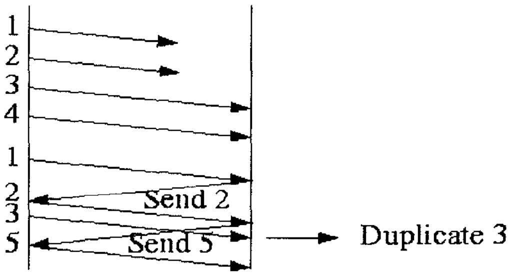


Figure 2: Timeout and Duplicate Packets in Slow Start

3 ABR Traffic Management

The ABR service uses a closed-loop rate-based traffic management model. We describe the relevant parts of the model briefly in this section. The Source End System (SES) is allowed to send data at the Allowed Cell Rate (ACR) which is less than a negotiated Peak Cell Rate (PCR). Immediately after establishing a connection, ACR is set to an Initial Cell Rate (ICR), which is also negotiated with the network. The source sends a Resource Management (RM) cell after transmitting every Nrm-1 cells (default Nrm value is 32). The RM cell contains a Current Cell Rate (CCR) field initialized with the current ACR, an Explicit Rate (ER) field where the network gives its feedback and a few other fields that are not relevant to our discussion here.

Based on their load, switches set the ER field in the RM cell. The RM cells return to the source carrying the minimum value of ER set by all the switches on the path. When the source receives the RM cell (feedback) from the network, it adjusts its ACR to the ER value in the RM cell. When there is a steady flow of RM cells in the forward and reverse directions, there is a steady flow of feedback from the network. In this state, we say that the ABR control loop has been established and the source rates are primarily controlled by the network feedback (closed-loop control).

When the source transmits data after an idle period, there is no reliable feedback from the network and, hence for one round-trip, the source rates are primarily controlled by the ABR source rules (open-loop control). The open-loop control is replaced by the closed-loop control once the control loop is established. Since bursty traffic consists of busy and idle periods, open-loop control may be exercised at the beginning of every burst. Hence, the source rules assume considerable importance in ABR flow control.

3.1 ABR Source End System Rules

One of the key parameters of this study was "transient buffer exposure" (TBE), which is used to limit the number of cells outstanding in the network. This in turn limits the number of cells lost if any link or switch in the path fails. The standard requires that the sources reduce their ACR if they have sent TBE cells and have not received a feedback from the network. The goal of this rule (known as source rule 6) is to avoid cell loss in the open-loop phase. It turns out that during closed-loop phase the number of cells in the network can be much larger and so the number of cells that may be lost due to congestion can be several order of magnitude larger than TBE. Rule 6 is rarely triggered during closed loop phase, even if the traffic is bursty.

The probability of Rule 6 triggering at the end of the burst is increased by decreasing the value of TBE. Though lower TBE may reduce instantaneous throughput by limiting the size of the burst, it may improve throughput since the time wasted in retransmissions is saved. This effect is seen only for very low values of TBE. On the other hand, TBE reduces ACR exponentially, and may rate-limit the TCP source, leading to drop in throughput.

3.2 The ERICA Switch Scheme

In this section, we present a brief overview of the ERICA switch algorithm. More details can be found in [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 Rate

Since VBR and CBR are serviced first, bandwidth available for ABR service class is given by:

ABR = Target Rate - VBR - CBR

The overload is measured with respect to the target rate (and not link rate):

Overload = Input Rate / ABR

In addition to the input rate, the switches also measure the number of active VCs and compute the fair share:

Fair Share = ABR / Number of Active VCs

For each VC, its share is computed based on the overload factor and the VC's current cell rate:

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

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

ER for VC = max (Fair Share, 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. We have enhanced this algorithm to reduce spiking effects due to transient overloads. We use the enhanced version in our simulation. The essential aspects of this study however remain the same for both versions of ERICA.

4 Source Model and TCP Options

We use an infinite source model at the application layer running on top of TCP. This implies that TCP always has a packet to send as long as its window will permit it. Other parameters values used are:

TCP maximum segment size MSS=512 bytes
IP MTU size = 9180 bytes (no IP segmentation)
TCP timer granularity = 100 ms
Delay-ack timer=0 (disabled)
Packet processing time at the destination=0

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 window of 16 × 64 kB or 1024 kB. The network consists of three links of 1000 km each and therefore has a one-way delay of 15 ms (or 291 kB at 155 Mbps). In our simulations, we have not used "fast retransmit and recovery" which is a subject for future study.

5 ABR Source End System and ERICA Parameters

The source end system parameters [2, 8] of ABR are selected to maximize the responsiveness and throughput. The values of source parameters are:

TBE = 128, 512
ICR = 10 Mbps
ADTF = 0.5 sec
CDF (XDF) = 0.5, CRM (Xrm) = TBE/Nrm
PCR = 155.52 Mbps, MCR = 0, RIF (AIR) = 1
Nrm = 32, Mrm = 2, RDF = 1/512,
Trm = 100 ms, TCR = 10 c/s

The ERICA switch algorithm parameters are chosen as follows. The target utilization parameter is chosen to be 90%. The overload and ABR capacity are measured at the switch over an interval of 100 cells or 1 ms (whichever is smaller). The buffer size at the bottleneck link is sized as TBE × n × 1, 2, or 4, where n is the number of ABR sources.

6 The n Source + VBR Configuration

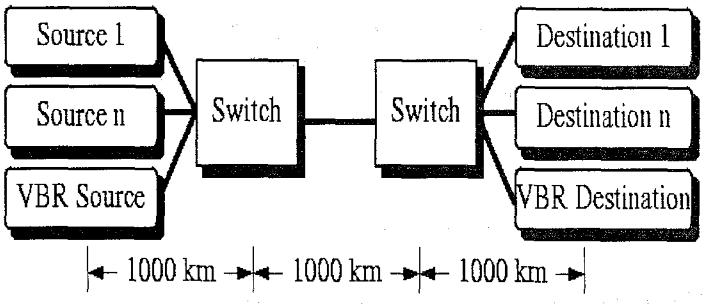


Figure 3: n Source + VBR Configuration

Figure 3 illustrates the general configuration we analyze, which we call "the n Source + VBR configuration." This configuration has a single bottleneck link between two switches. The link capacity is shared by n ABR sources and possibly a VBR source. All links run at 155 Mbps and are 1000 km long.

The VBR background is optional. When present, it is an ON-OFF source with a 100 ms ON time and 100 ms OFF time. The VBR starts at t = 2 ms to avoid certain initialization problems. The maximum amplitude of the VBR source is 124.41 Mbps (80% of link rate). This is deliberately set below the ERICA target utilization of 90%. By doing so, we always leaves at least 10% for ABR. This avoids scheduling issues. We may safely assume that VBR is given priority at the link, i.e, if there is a VBR cell, it will be scheduled for output on the link before any waiting ABR cells are scheduled. Also, since ABR bandwidth is always non-zero, the ABR sources are never allocated zero rates. We, thus, avoid the need for out-of-rate RM cells, which are required if an ABR source is allocated an ACR of zero and cannot send any data cells.

All traffic is unidirectional. A large (infinite) file transfer application runs on top of TCP for the TCP sources. We experiment with 2 values of $n = 2$ and 5. The buffer size at the bottleneck link is sized as TBE $\times n \times$ {1, 2, or 4}.

7 Performance Metrics

We measure throughput of each source and cell loss ratio. Also, we can plot a number of variables as a function of time that help explain the behavior of the system. These include TCP sequence numbers at the source, congestion window (CWND), ACR of each source, link utilization, and queue length.

We define TCP throughput as the number of bytes delivered to the destination application in the total time. This is sometimes referred to as goodput by other authors. Cell Loss Ratio (CLR) is measured as the ratio of the number of cells dropped to the number of cells sent during the simulation.

The following equation should hold for the aggregate metrics of the simulation:

Number of bytes sent = Bytes sent once
+ Bytes retransmitted
= Bytes delivered to application
+ Data bytes dropped at the switch + Bytes in the path
+ Partial packet bytes dropped at the destination AAL5
+ Duplicate packet bytes dropped at the destination TCP

The places where cells or packets are dropped are illustrated in Figure 4.

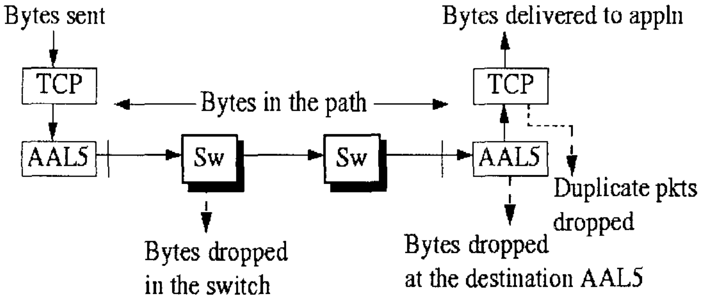


Figure 4: Cell/Packet Drop Points on a TCP/ATM connection

8 Peak TCP Throughput

In order to measure the best possible throughput of TCP over ABR, we first present the results of a case with infinite buffers and fixed ABR capacity. With finite buffers or variable ABR capacity, it is possible that some cells are lost, which may result in unnecessary timeouts and retransmissions leading to reduced throughput. Fixed ABR capacity is achieved by not having any VBR source in this case.

We simulate the configuration with $n = 2$, buffer size = 4096 and TBE = 512. In this case, no cells are lost, the CLR is zero and the throughput is 103.32 Mbps. This is the maximum TCP throughput with two sources in this configuration. It can be approximately verified as follows:

Throughput = 155 Mbps
× 0.9 for ERICA Target Utilization
× 48/53 for ATM payload
× 512/568 for protocol headers
(20 TCP + 20 IP + 8 RFC1577 + 8 AAL5 = 56 bytes)
× 31/32 for ABR RM cell overhead
× a fraction (0.9) to account for the TCP startup time
≃ 103.32 Mbps

Figure 5 shows graphs of window size, sequence numbers, and ACR for the two sources. Note that the curves for the two sources completely overlap indicating that the performance is fair. Also, the sources use the entire ACR allocated to them. In other words,*the TCP sources are rate-limited and not window-limited.*

9 Effect of Finite Buffers

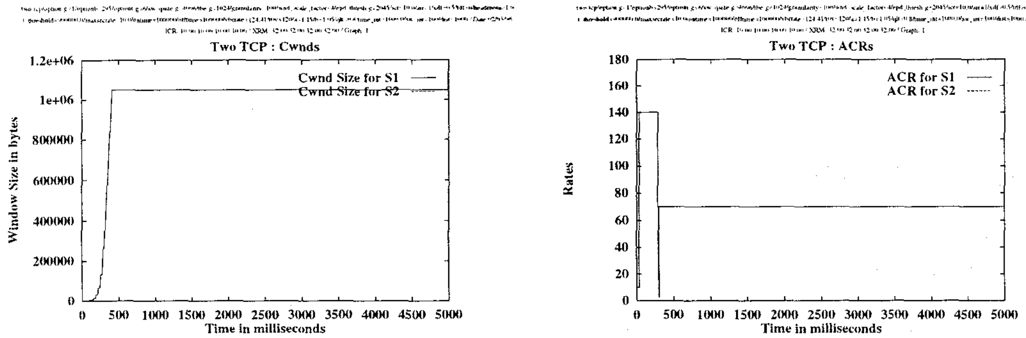
We now investigate the effect of smaller buffers, keeping the ABR capacity fixed. The buffer size is set to the product of TBE (512), the number of sources (2), and a safety factor (2), i.e., 2048 = 512 × 2 × 2. The remaining configuration is the same as in Section 8 i.e., $n = 2$, TBE = 512 and fixed ABR capacity (no VBR source). Since the buffers are smaller, it is possible that they might overflow before the ABR control loop is set up. We expect some cell loss and reduced throughput due to timeout retransmission.

We observe that there is a drastic reduction of TCP throughput which is not proportional to the increase in CLR. The throughput drops by 36% while the CLR is only 0.18%.

Figure 6 shows graphs of window size, sequence numbers, and ACR for the two sources. Figure 6(a) shows that there is one loss around t=200 ms. No acks are received for the next 300 ms and therefore, the window remains constant and finally drops to 1 at t=500 ms. The packets are retransmitted and window rises exponentially upto the half of the value before the drop. Subsequently, the window rise linearly. Note that the linear rise is very slow. The source window is much below its maximum. In other words, the sources are window limited. The congestion windows of both sources are approximately equal, and so the operation is fair. However, the throughput in this experiment is only 64% of the maximum throughput. The measured cell loss ratio in this case was only 0.18%. Note that the *CLR and throughput loss are one order of magnitude apart.*

Figure 6(b) shows the rates (ACRs) allocated to the two sources. Notice that the curves for the two sources and are at the maximum possible value (90% of the link rate) and so the sources have a large ACR. The reason for throughput being less than maximum possible is not the sources' ACRs but their windows. That is,*the sources are not rate-limited but are window-limited.* Also, notice that the two curves overlap. This shows that the ABR rate allocation mechanism is fair.

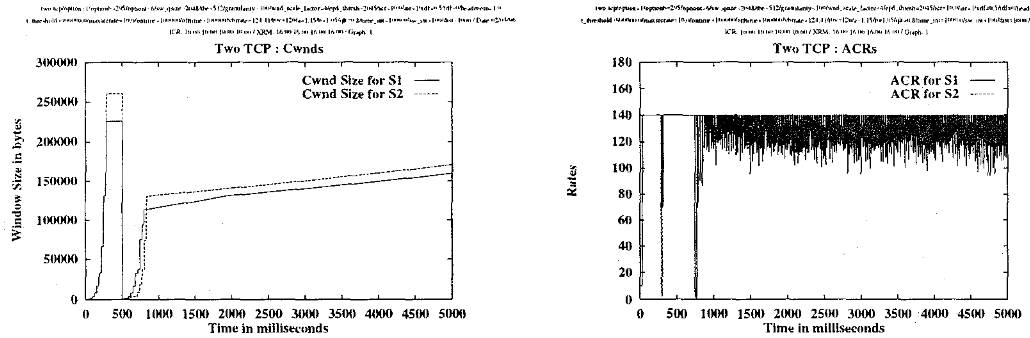
The main reason for the large drop in throughput is that cells (packets) are dropped. Each cell loss results in a significant loss of time and throughput. In this case, this happens before the ABR control loop is set up (open-loop period). The TBE



(a) Congestion Window

(b) ACR

Figure 5: Two TCP Source Configuration, Buffer=4096 cells, TBE=1024



(a) Congestion Window

(b) ACR

Figure 6: Two TCP Source Configuration, Buffer=2048 cells, TBE=512

in this case was 512. For two sources, one would assume that having 1024 buffers in the switch would be sufficient. But this case shows that cells are lost even when there are twice as many (2048) buffers in the switch. Thus, *TBE is not a good mechanism to control or allocate buffers.* This observation was also made in our earlier work on non-TCP bursty traffic [9, 5].

10 Effect of Finite Buffers and Varying ABR Capacity

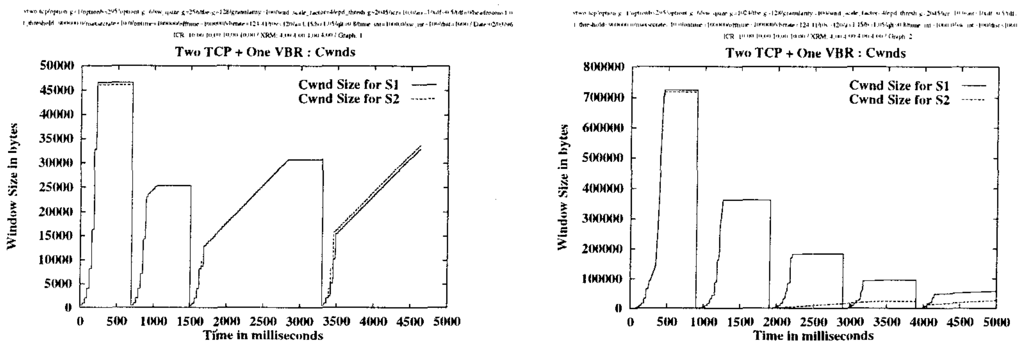
Next we studied the effect of varying ABR capacity. For this purpose, we introduce a VBR traffic in the background. We conducted several experiments with two and five ABR sources. Since VBR takes up 40% of the link bandwidth, we expect the maximum ABR throughput to be 60% of the case without VBR.

Figures 7 and 8 show the window graphs for the two- and five-source source configurations, respectively. Four different TBE and buffer size combinations are used. The graphs clearly show the instants when cells are lost and the TCP windows adjusted. The ACR and sequence number graphs have not been included here since there is not much new information in them.

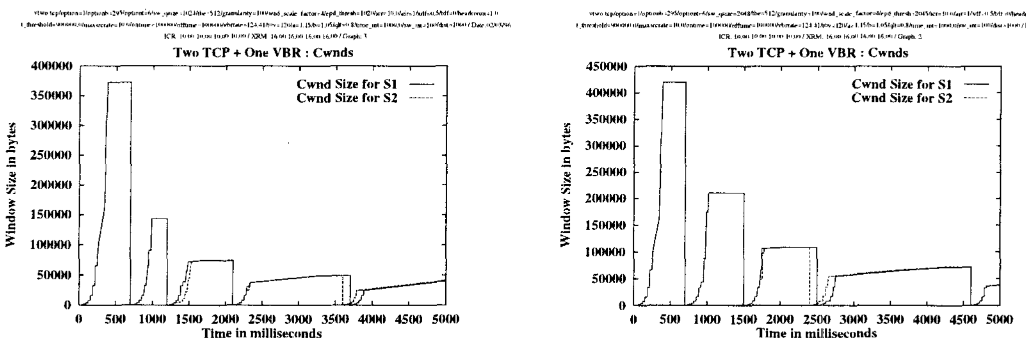
The simulation results are summarized in Table 1 and are discussed in the following subsection. The first column is the configuration used. The second and third columns show the TBE and the buffer sizes used. T1 through T5 are the throughput values for sources 1 through 5. We also show the total ABR throughput. It is helpful to express it as a percentage of maximum possible ABR throughput (58.4 Mbps). The last column shows the CLR.

From this table we can make the following conclusions:

1. CLR vs Throughput: Table 1 shows that that *CLR is small and has high variance.* CLR does not reflect TCP performance since higher CLR does not necessarily mean lower TCP throughput. The effect of cell loss depends not upon the number of cells lost but upon the the number of timeouts. If a large number of cells are lost but there is only one timeout, the throughput degradation may not be that severe. On the other hand, even if a few cells are lost but the losses happen far apart triggering multiple timeouts, the throughput will be severely degraded. Hence, the cell level metric *CLR is not a good indicator of the TCP level performance.*



(a) buffer=256, TBE=128, Congestion Window (b) buffer=1024, TBE=128, Congestion Window



(c) buffer=1024, TBE=512, Congestion Window (d) buffer=2048, TBE=512, Congestion Window

Figure 7: Two TCP + One VBR Configuration, TBE vs Buffer

2. Effect of Buffering: *Larger buffers always give higher TCP throughput* for our infinite TCP applications.

   We have not studied the packet latency. It has been claimed that too many buffers may increase latency for client-server applications. However, we have not verified or disproved that claim.

   The effect of large buffers on CLR is mixed. With large buffering, windows can be large and if the a loss occurs at a large window, CLR can be high. On the other hand, if the loss occurs at a low window, CLR can be low.

3. Effect of Multiple Sources: *As the number of sources is increased, generally the total throughput increases.* This is because, these TCP sources are generally window limited and five sources with small windows pump more data than two sources with small windows.

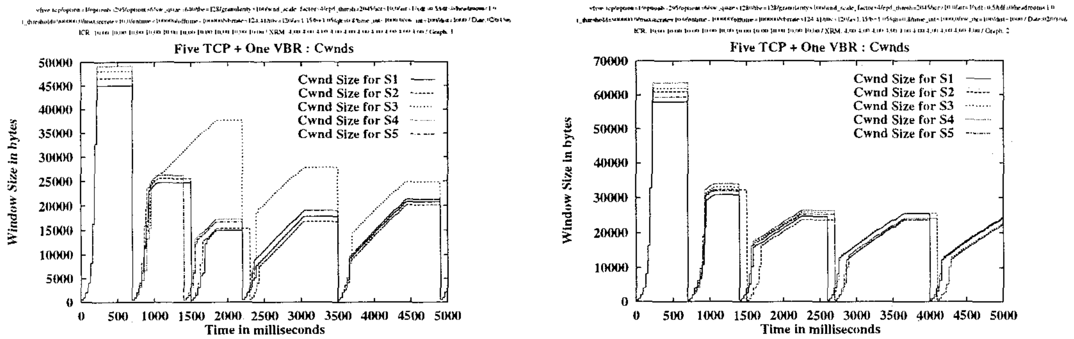
11 Observations on Tail Drop

In this section we report an interesting phenomenon due to tail drop and propose a simple fix. In AAL5, the source marks the last cell of each message by End-of-Message (EOM) bit. If the EOM cell is dropped at the switch, the retransmitted packet gets merged with previous partial packet at the destination. The merged packet fails the CRC test and is dropped at the destination by AAL5. The source will have to retransmit two packets.

After the first retransmission, the SSTHRESH is set to half the previous window size and the window is set to one. When the second retransmission occurs, the window is one and hence SSTHRESH is set to 2 (the minimum value). The window remains at one. TCP henceforth increases the window linearly resulting in low throughput for this source. Since the EOM cells of the other TCP sources may not have been dropped, they do not experience this phenomenon and get high throughput.

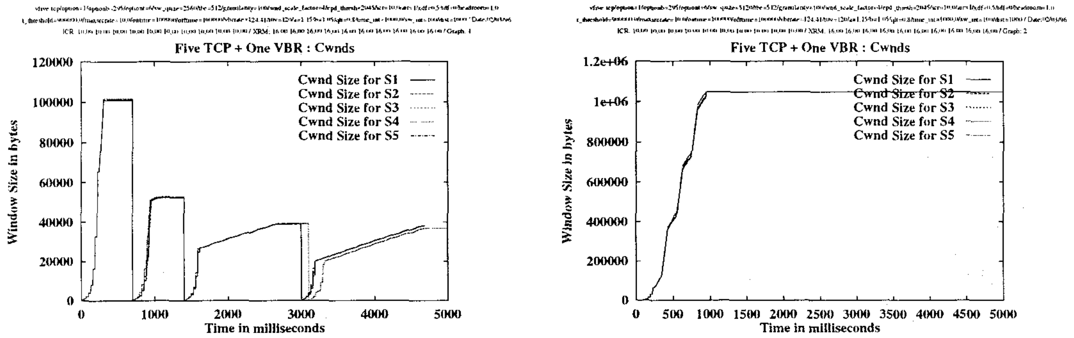
The disparity in throughput results in unfairness among sources as shown in Figure 9. Figure 7(b) shows a simulation where this unfairness is seen. In this figure, source S2 loses cells at 400 ms and 1300 ms. The corresponding timeout and retransmissions occur at 900 ms and 1900 ms. The merging of the packets at the AAL5 occurs at 1300 ms. After the second timeout, the window of S2 increases linearly from one. Since source S1 does not experience this phenomenon, it gets higher throughput.

A simple fix is what we call Intelligent Tail Drop. This policy sets a threshold a few cells before the buffer limit. Once the threshold is crossed, the switch drops all cells *except the EOM cells.* The EOM cells will reach the destination and result in the dropping of the first packet and *merging of packets is*



(a) buffer=640, TBE=128, Congestion Window

(b) buffer=1280, TBE=128, Congestion Window



(c) buffer=2560, TBE=512, Congestion Window

(d) buffer=5120, TBE=512, Congestion Window

Figure 8: Five TCP + One VBR Configuration, TBE vs Buffer

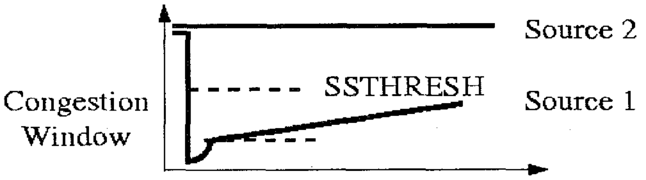


Figure 9: Unfairness due to TailDrop

avoided in the destination AAL5. This prevents the back-to-back retransmissions and *improves fairness.* Since this policy only enhances tail drop, it can still be used in conjunction with other drop policies like Early Packet Discard (EPD) [14]. A similar policy for partial packet discard is described in Reference [1].

12 Summary

We have studied the effect of running TCP/IP traffic with ABR. The main results of the study are:

1. TCP achieves maximum throughput when there are enough buffers at the switches.
2. When maximum throughput is achieved, the TCP sources are rate-limited by ABR rather than window-limited by TCP.
3. When the number of buffers is smaller, there is a large reduction in throughput even though CLR is very small.
4. The reduction in throughput is due to loss of time during timeouts (large timer granularity), and transmission of duplicate packets which are dropped at the destination.
5. When throughput is reduced, the TCP sources are window-limited by TCP rather than rate-limited by ABR.
6. Switch buffers should not be dimensioned based on the ABR Source parameter TBE. In our later studies [10, 12], we show that a switch with buffers equal to a small multiple of network diameter can guarantee no loss even for a very large number of VCs carrying TCP/IP traffic. In other words the ABR service is scalable in terms of the number of TCP/IP sources (or VCs).
7. When ABR capacity is varied, CLR exhibits high variance and is not related to TCP throughput. In general, CLR is not a good indicator of TCP level performance.
8. Larger buffers increase TCP throughput.
9. Larger number of window-limited sources increase TCP throughput. This is because, the sum of the windows is larger when there are more sources.

Table 1: Simulation Results: Summary

Number of			Throughput							
Sources	TBE	Buffer	T1	T2	T3	T4	T5	Total	% of Max	CLR
2 ABR + VBR	128	256	3.1	3.1				6.2	10.6	1.2
2 ABR + VBR	128	1024	10.5	4.1				14.6	24.9	2.0
2 ABR + VBR	512	1024	5.7	5.9				11.6	19.8	2.7
2 ABR + VBR	512	2048	8.0	8.0				16.0	27.4	1.0
5 ABR + VBR	128	640	1.5	1.4	3.0	1.6	1.6	9.1	15.6	4.8
5 ABR + VBR	128	1280	2.7	2.4	2.6	2.5	2.6	12.8	21.8	1.0
5 ABR + VBR	512	2560	4.0	4.0	4.0	3.9	4.1	19.9	34.1	0.3
5 ABR + VBR	512	5720	11.7	11.8	11.6	11.8	11.6	58.4	100.0	0.0

10. Even when the buffers are small, dropping of EOM cells should be avoided. This avoids merging of packets at the destination AAL5 and improves fairness.

11. ABR is better than UBR. Our subsequent studies [11] show that UBR switches require buffers proportional to the sum of the TCP receiver window sizes. In other words the UBR service is *not* scalable in terms of the number of TCP/IP sources (or VCs).

References

[1] G.J. Armitage, K.M. Adams, "ATM Adaptation Layer Packet Reassembly during Cell Loss", *IEEE Network*, September 1993.

[2] ATM Forum, "ATM Traffic Management Specification Version 4.0," April 1996, available as 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," *AF-TM 95-1645*,[1], December 1995.

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

[5] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, F. Lu, "On Source Rules For ABR Service on ATM Networks with Satellite Links," Submitted to the First International Workshop on Satellite-based Information Services (WOSBIS), New York, November 1996.

[6] V. Jacobson, "Congestion Avoidance and Control," *Proceedings of the SIGCOMM'88 Symposium*, pp. 314-32, August 1988.

[7] Raj Jain, Shiv Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Ram Viswanathan, "ERICA Switch Algorithm: A Complete Description," *AF-TM 96-1172*, August 1996.

[8] Raj Jain, Shiv Kalyanaraman, Rohit Goyal, Sonia Fahmy, "Source Behavior for ATM ABR Traffic Management: An Explanation," To appear in IEEE Communications Magazine, November 1996. [2]

[9] R.Jain, S.Fahmy, S.Kalyanaraman, R. Goyal, F.Lu, "More Strawvote Comments: TBE vs Queue sizes," *AF-TM 95-1661*, December 1995.

[10] Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Rohit Goyal, S.M.Srinidhi, "Buffer Requirements for TCP over ABR," *AF-TM 96-0517*, April 1996.

[11] Raj Jain, Rohit Goyal, Shiv Kalyanaraman, Sonia Fahmy, S.M.Srinidhi, "Performance of TCP/IP over UBR and buffer requirements," *AF-TM 96-0518*, April 1996.

[12] Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, "Buffer Requirements For TCP/IP Over ABR," Proceedings of IEEE ATM 96 Workshop, San Fransisco, August 1996.

[13] Hongqing Li, Kai-Yeung Siu, Hong-Ti Tzeng, Chinatsu Ikeda and Hiroshi Suzuki "TCP over ABR and UBR Services in ATM," *Proceedings of IPCCC'96*, March 1996.

[14] 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